Number Systems

Number System

- The technique to represent and work with numbers is called **number system**.
- **Decimal number system** is the most common number system.
- Other popular number systems include **binary number system, octal number system, hexadecimal number system,** etc.
- Computers cannot understand human languages, so **to understand the commands and instructions given to the computers by programmers**, different number systems are used such as binary, octal, decimal, etc.
- **Types of Number Systems**
- There are different types of number systems in which the four main types are as follows.
    1. Binary number system (Base - 2)
    2. Octal number system (Base - 8)
    3. Decimal number system (Base - 10)
    4. Hexadecimal number system (Base - 16)

Quantities/Counting (1 of 3)

Decimal	Binary	Octal	Hexa-decimal
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7

Quantities/Counting (2 of 3)

Decimal	Binary	Octal	Hexa-decimal
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

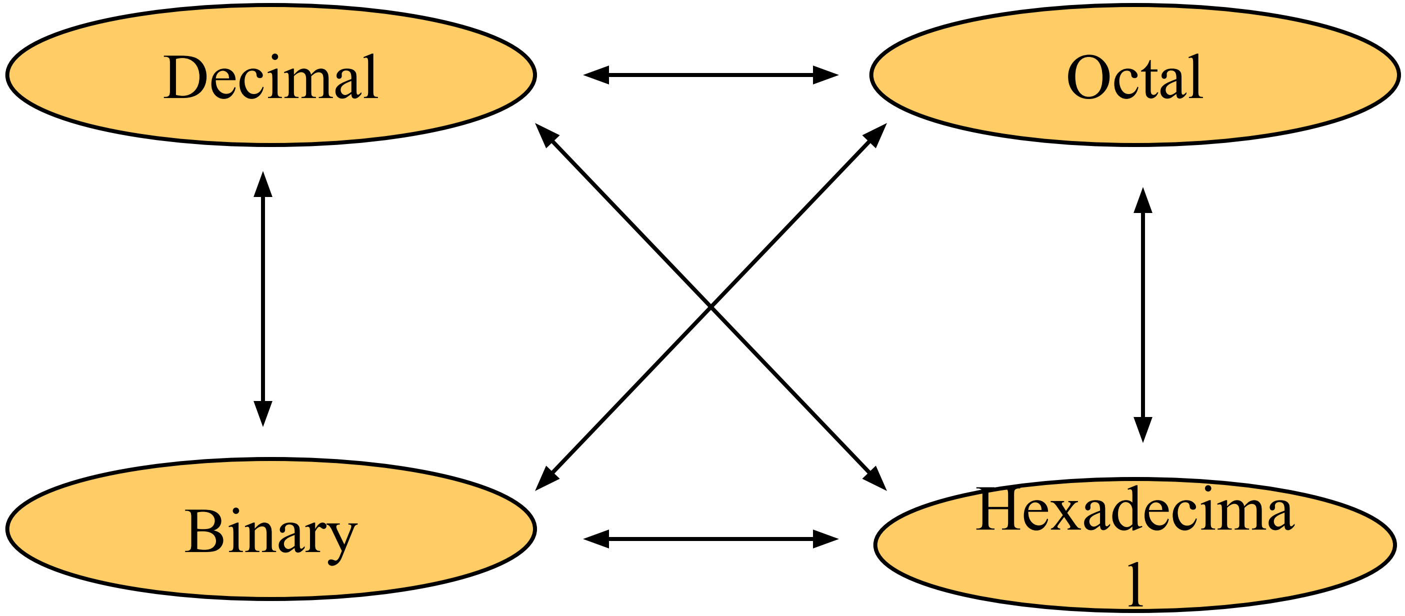
Quantities/Counting (3 of 3)

Decimal	Binary	Octal	Hexa-decimal
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17

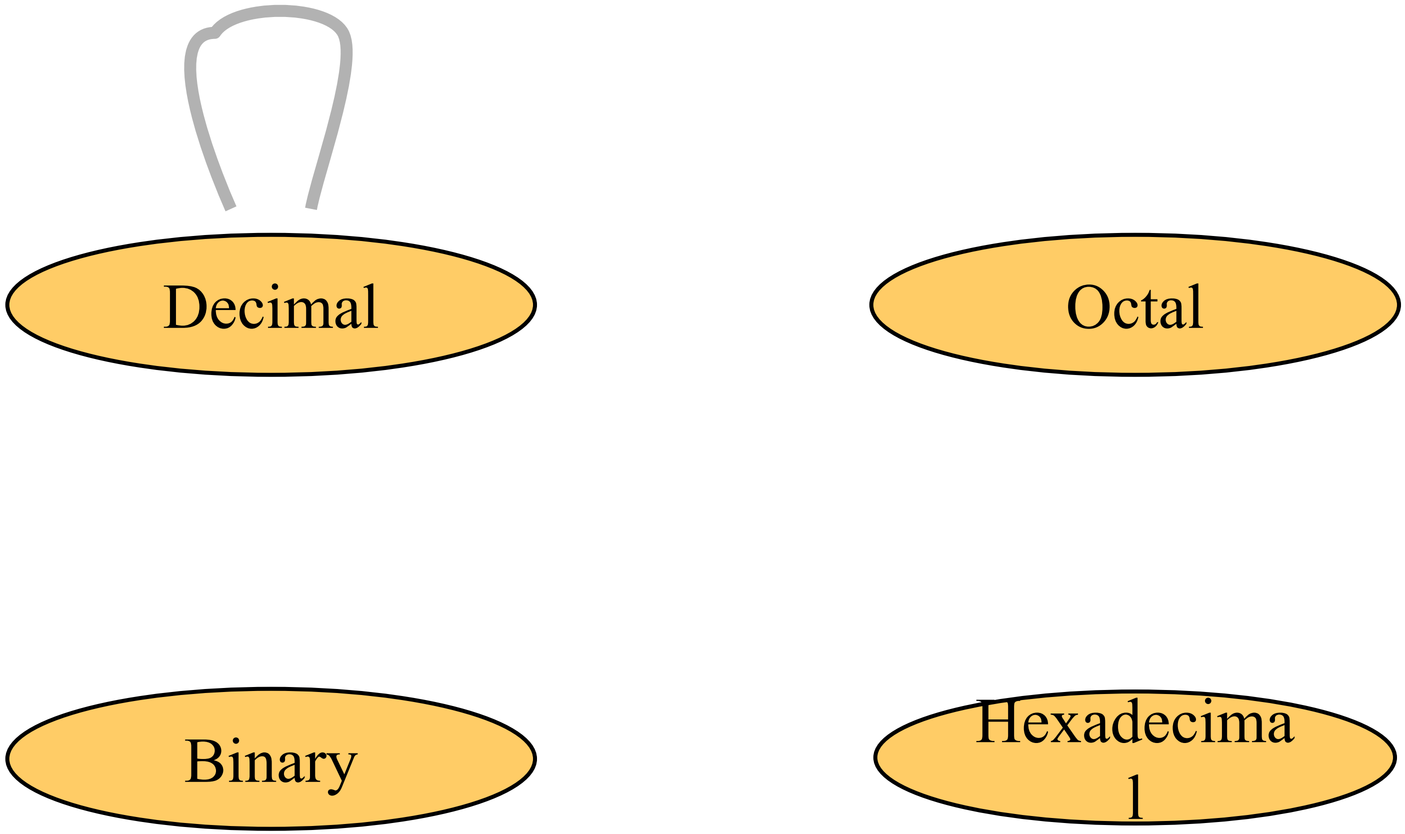
Etc.

Conversion Among Bases

- The possibilities:



Decimal to Decimal



Next slide…

Weight

125_{10} => $5 \times 10^{0} = 5$

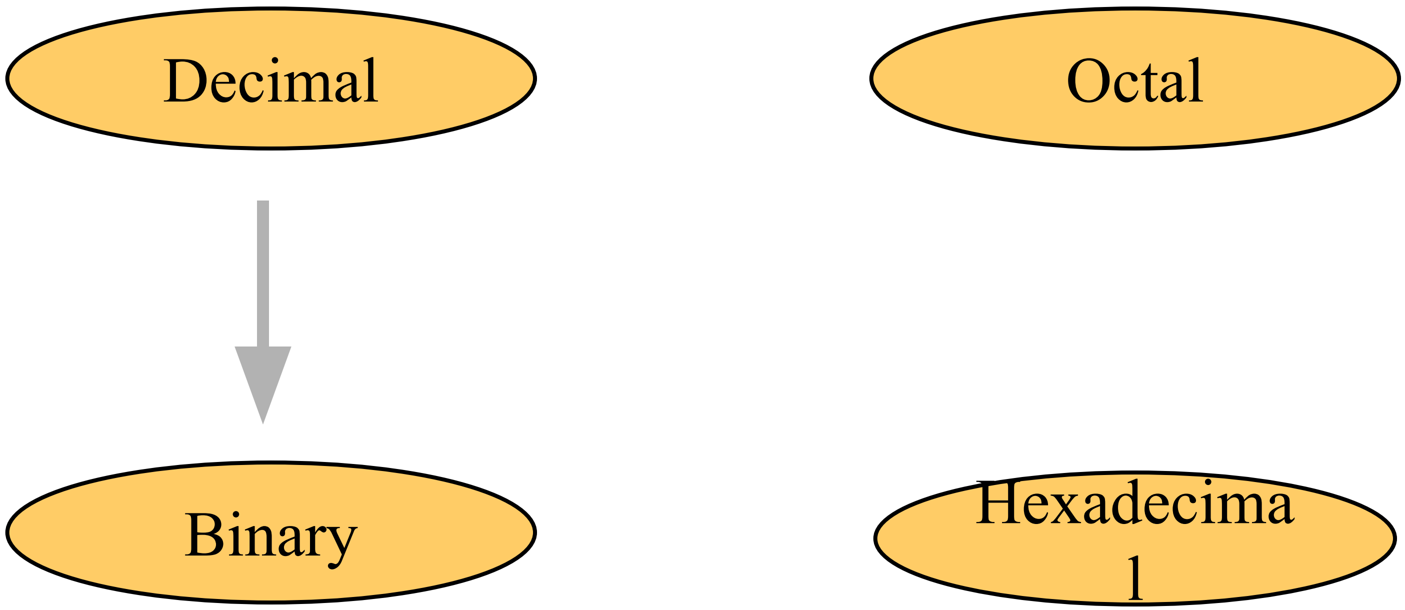
$2 \times 10^{1} = 20$

$1 \times 10^{2} = 100$

125

Base

Decimal to Binary



Decimal to Binary

- Technique
  - Divide by two, keep track of the remainder

Example

$125_{10} = ?_2$

2	125	
2	62	1
2	31	0
2	15	1
2	7	1
2	3	1
2	1	1
	0	1

$125_{10} = 1111101_2$

Convert 56_{10} into a binary number.

$56_{10} = ?_2$

Division by 2	Quotient	Remainder	Binary Bit
56 ÷ 2	28	0	0(LSB)
28 ÷ 2	14	0	0
14 ÷ 2	7	0	0
7 ÷ 2	3	1	1
3 ÷ 2	1	1	1
1 ÷ 2	0	1	1

$\therefore (56)_{10} = (111000)_2$

Convert 278_{10} into a binary number.

$278_{10} = ?_2$

Division by 2	Quotient	Remainder	Binary Bit
278 ÷ 2	139	0	0 (LSB)
139 ÷ 2	69	1	1
69 ÷ 2	34	1	1
34 ÷ 2	17	0	0
17 ÷ 2	8	1	1
8 ÷ 2	4	0	0
4 ÷ 2	2	0	0
2 ÷ 2	1	0	0
1 ÷ 2	0	1	1 (MSB)

Solution:
$\therefore (278)_{10} = (100010110)_2$

Convert 180_{10} into a binary number.

$180_{10} = ?_2$

Division by 2	Quotient	Remainder	Binary Bit
180 ÷ 2	90	0	0(LSB)
90 ÷ 2	45	0	0
45 ÷ 2	22	1	1
22 ÷ 2	11	0	0
11 ÷ 2	5	1	1
5 ÷ 2	2	1	1
2 ÷ 2	1	0	0
1 ÷ 2	0	1	1(MSB)

$\therefore (180)_{10} = (10110100)_2$

Convert 1073_{10} into a binary number.

$1073_{10} = ?_2$

Division by 2	Quotient	Remainder	Binary Bit
1073 ÷ 2	536	1	1 (LSB)
536 ÷ 2	268	0	0
268 ÷ 2	134	0	0
134 ÷ 2	67	0	0
67 ÷ 2	33	1	1
33 ÷ 2	16	1	1
16 ÷ 2	8	0	0
8 ÷ 2	4	0	0
4 ÷ 2	2	0	0
2 ÷ 2	1	0	0
1 ÷ 2	0	1	1

$\therefore (1073)_{10} = (10000110001)_2$

Convert 81_{10} into a binary number.

$81_{10} = ?_2$

Division by 2	Quotient	Remainder	Binary Bit
81 ÷ 2	40	1	1(LSB)
40 ÷ 2	20	0	0
20 ÷ 2	10	0	0
10 ÷ 2	5	0	0
5 ÷ 2	2	1	1
2 ÷ 2	1	0	0
1 ÷ 2	0	1	1(MSB)

$\therefore (81)_{10} = (1010001)_2$

Convert 403_{10} into a binary number.

$403_{10} = ?_2$

Division by 2	Quotient	Remainder	Binary Bit
403 ÷ 2	201	1	1(LSB)
201 ÷ 2	100	1	1
100 ÷ 2	50	0	0
50 ÷ 2	25	0	0
25 ÷ 2	12	1	1
12 ÷ 2	6	0	0
6 ÷ 2	3	0	0
3 ÷ 2	1	1	1
1 ÷ 2	0	1	1(MSB)

$\therefore (403)_{10} = (110010011)_2$

Convert 508_{10} into a binary number.

$508_{10} = ?_2$

Division by 2	Quotient	Remainder	Binary Bit
508 ÷ 2	254	0	0(LSB)
254 ÷ 2	127	0	0
127 ÷ 2	63	1	1
63 ÷ 2	31	1	1
31 ÷ 2	15	1	1
15 ÷ 2	7	1	1
7 ÷ 2	3	1	1
3 ÷ 2	1	1	1
1 ÷ 2	0	1	1(MSB)

$\therefore (508)_{10} = (111111100)_2$

Convert 1278_{10} into a binary number.

$1278_{10} = ?_2$

Division by 2	Quotient	Remainder	Binary Bit
1278 ÷ 2	639	0	0(LSB)
639 ÷ 2	319	1	1
319 ÷ 2	159	1	1
159 ÷ 2	79	1	1
79 ÷ 2	39	1	1
39 ÷ 2	19	1	1
19 ÷ 2	9	1	1
9 ÷ 2	4	1	1
4 ÷ 2	2	0	0
2 ÷ 2	1	0	0
1 ÷ 2	0	1	1(MSB)

$\therefore (1278)_{10} = (10011111110)_2$

Convert 145_{10} into a binary number.

$145_{10} = ?_2$

Division by 2	Quotient	Remainder	Binary Bit
145 ÷ 2	72	1	1 (LSB)
72 ÷ 2	36	0	0
36 ÷ 2	18	0	0
18 ÷ 2	9	0	0
9 ÷ 2	4	1	1
4 ÷ 2	2	0	0
2 ÷ 2	1	0	0
1 ÷ 2	0	1	1 (MSB)

$\therefore (145)_{10} = (10010001)_2$

Practice Questions

Practice Questions on Decimal to Binary Conversions

1. Convert 155_{10} into a binary number.
2. Convert 375_{10} into a binary number.
3. Convert 74_{10} into a binary number.

Decimal(with fraction) to Binary

Example : 4.47

Step 1: Conversion of 4 to binary

```
1.   4/2 : Remainder = 0 : Quotient = 2
2.   2/2 : Remainder = 0 : Quotient = 1
3.   1/2 : Remainder = 1 : Quotient = 0
```

So equivalent binary of integral part of decimal is 100.

Step 2: Conversion of .47 to binary

```
1.    0.47 * 2 = 0.94, Integral part: 0
2.    0.94 * 2 = 1.88, Integral part: 1
3.    0.88 * 2 = 1.76, Integral part: 1
```

*So equivalent binary of fractional part of decimal is .011 (*Repeat steps until you either get to 0 or a pe

Step 3: Combined the result of step 1 and 2.

Final answer can be written as: 100.011

Decimal(with fraction) to Binary

Example : 8.125

Step 1: Conversion of 8 to binary

1. 8/2 : Remainder = 0 : Quotient = 4
2. 4/2 : Remainder = 0 : Quotient = 2
3. 2/2 : Remainder = 0 : Quotient = 1
4. 1/2 : Remainder = 1 : Quotient = 0

So equivalent binary of integral part of decimal is 1000.

Step 2: Conversion of .125 to binary

1. 0.125 * 2 = 0.250, Integral part: 0
2. 0.250 * 2 = 0.500, Integral part: 0
3. 0.500 * 2 = 1.000, Integral part: 1

So equivalent binary of fractional part of decimal is .001

Step 3: Combined the result of step 1 and 2.

Final answer can be written as: 1000.001

Fractions

- Decimal to binary
  - Divide by 2, keep track of reminder

3.14579

```
 .14579
x     2
-------
0.29158
x     2
-------
0.58316
x     2
-------
1.16632
x     2
-------
0.33264
x     2
-------
0.66528
x     2
-------
1.33056
etc.
```

11.001001...

Fractions

- Decimal to binary

45.25

```
 .25
 x 2
-----
 0.5
 x 2
-----
 1.0
```

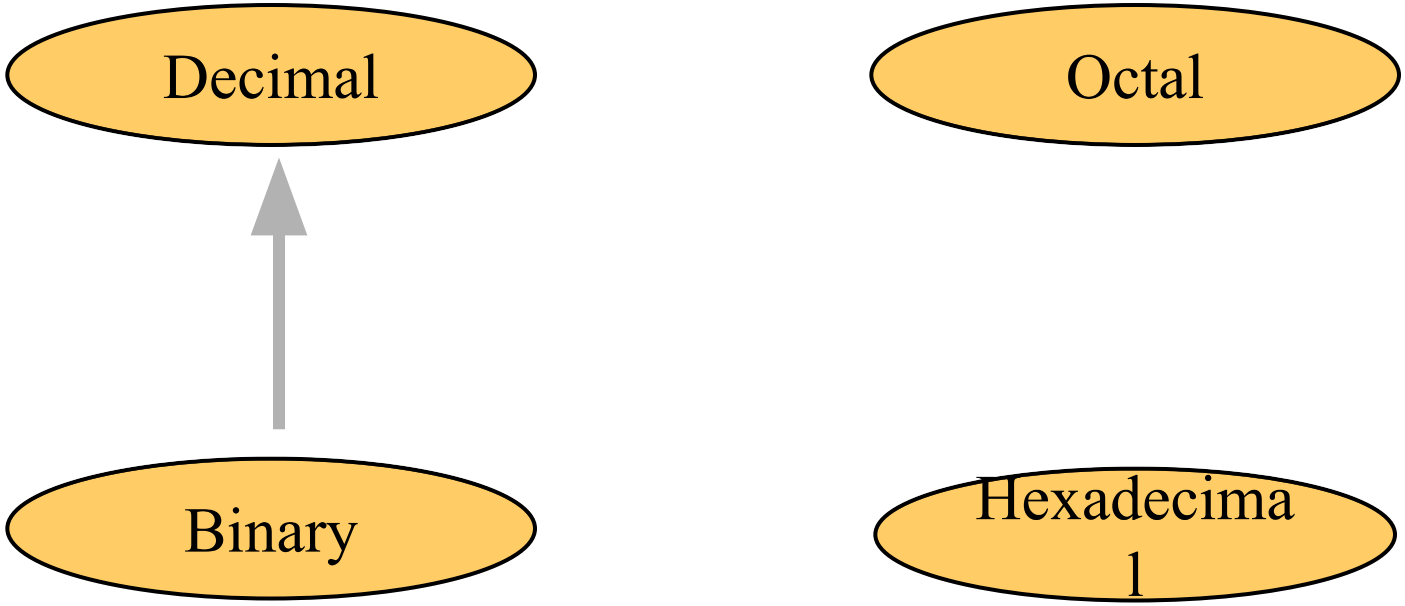
101101.01

Fractions

- Decimal to binary

  `15.6` = **(1111.*1001*)**

Binary to Decimal



Binary to Decimal

- Technique
  - Multiply each bit by 2^n, where n is the “weight” of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

Bit "0"

$$101011_2 \Rightarrow \begin{array}{rcl} 1 \times 2^0 & = & 1 \\ 1 \times 2^1 & = & 2 \\ 0 \times 2^2 & = & 0 \\ 1 \times 2^3 & = & 8 \\ 0 \times 2^4 & = & 0 \\ 1 \times 2^5 & = & \underline{32} \\ & & 43_{10} \end{array}$$

Convert 0110101 to decimal.

Given Binary number is 0110101

0110101 = $(0x2^6) + (1x2^5) + (1x2^4) + (0x2^3) + (1x2^2) + (0x2^1) + (1x2^0)$
= 0 + 32 + 16 + 0 + 4 + 0 + 1
= 53
Therefore, Binary Number 0110101 = 53 Decimal number

Convert the binary number 10100011 to decimal.

Given binary number is 10100011

Using the conversion formula,

10100011 = $(1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$
$= 128 + 0 + 32 + 0 + 0 + 0 + 2 + 1$
$= 163$

Therefore, binary number 10100011 = 163 decimal number

Convert the binary number 11101111 to decimal.

Given binary number is 11101111

Using the conversion formula,

11101111 $= (1 \times 2^7) + (1 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$
$= 128 + 64 + 32 + 0 + 8 + 4 + 2 + 1$
$= 239$

Therefore, binary number 11101111 = 239 decimal number

Convert the binary number 1001 to a decimal number.

Given, binary number = 1001_2

Hence, using the binary to decimal conversion formula, we have:

$1001_2 = (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$
$= 8 + 0 + 0 + 1$
$= (9)_{10}$

Convert 1101001_2 into an equivalent decimal number.

Solution: Using binary to decimal conversion method, we get;

$(1101001)_2 = (1 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

$= 64 + 32 + 0 + 8 + 0 + 0 + 1$

$= (105)_{10}$

Convert $(11110111)_2$ into base-10 number system.

Solution: Using binary to decimal conversion method, we get;

$(11110111)_2 = (1 \times 2^7) + (1 \times 2^6) + (1 \times 2^5) + (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$= 128 + 64 + 32 + 16 + 0 + 4 + 2 + 1$

$= (247)_{10}$

Convert $(110.101)_2$ into base-10 number system.

Step 1: Conversion of 110 to decimal

$=> 110_2 = (1*2^2) + (1*2^1) + (0*2^0)$

$\Rightarrow 110_2 = 4 + 2 + 0 => 110_2 = 6$

$\Rightarrow$ *So equivalent decimal of binary integral is 6.*

Step 2: Conversion of .101 to decimal

$0.101_2 = (1*2^{-1}) + (0*2^{-2}) + (1*2^{-3})$

$\Rightarrow 0.101_2 = (1*1/2) + (0*1/2^2) + (1*1/2^3)$

$\Rightarrow 0.101_2 = 1*0.5 + 0*0.25 + 1*0.125$

$\Rightarrow 0.101_2 = 0.625$

$\Rightarrow$ *So equivalent decimal of binary fractional is 0.625*

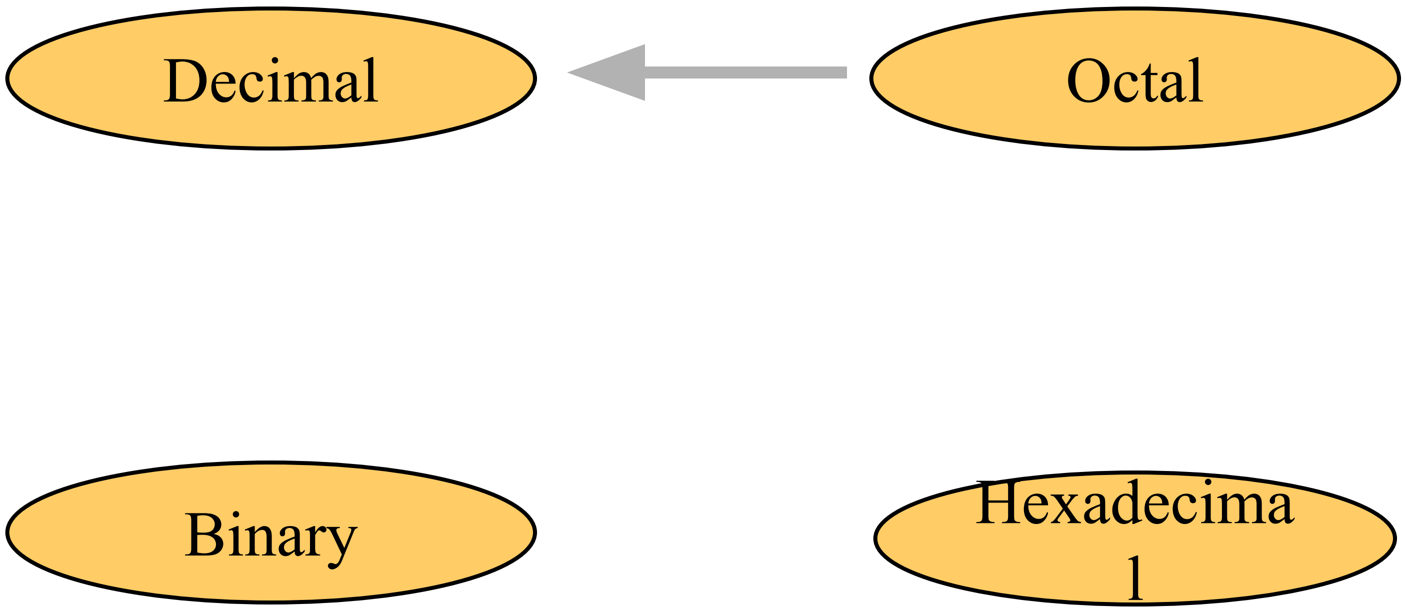
Step 3: Add result of step 1 and 2.

=> 6 + 0.625 = 6.625

Practice Questions

Convert binary number 10111 into an equivalent decimal number.
Convert 111_2 in decimal number.
What is 101010_2 in decimal number?

Octal to Decimal



Octal to Decimal

- Technique
  - Multiply each bit by 8^n, where n is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Example

$$
\begin{aligned}
724_8 \Rightarrow \quad 4 \times 8^0 &= \quad 4 \\
2 \times 8^1 &= \quad 16 \\
7 \times 8^2 &= \quad 448 \\
& \quad 468_{10}
\end{aligned}
$$

Example

- **$(304)_8 = (\underline{\quad ? \quad})_{10}$**

$=3\times8^2+0\times8^1+4\times8^0$

$=3\times64+0\times8+4\times1$

$=196$

$\therefore (304)8=(196)10$

Example

- **$(1534)_8$ = $(\underline{\quad ? \quad})_{10}$**

$=1\times8^3+5\times8^2+3\times8^1+4\times8^0$

$=1\times512+5\times64+3\times8+4\times1$

$=860$

$\therefore (1534)8=(860)10$

Example

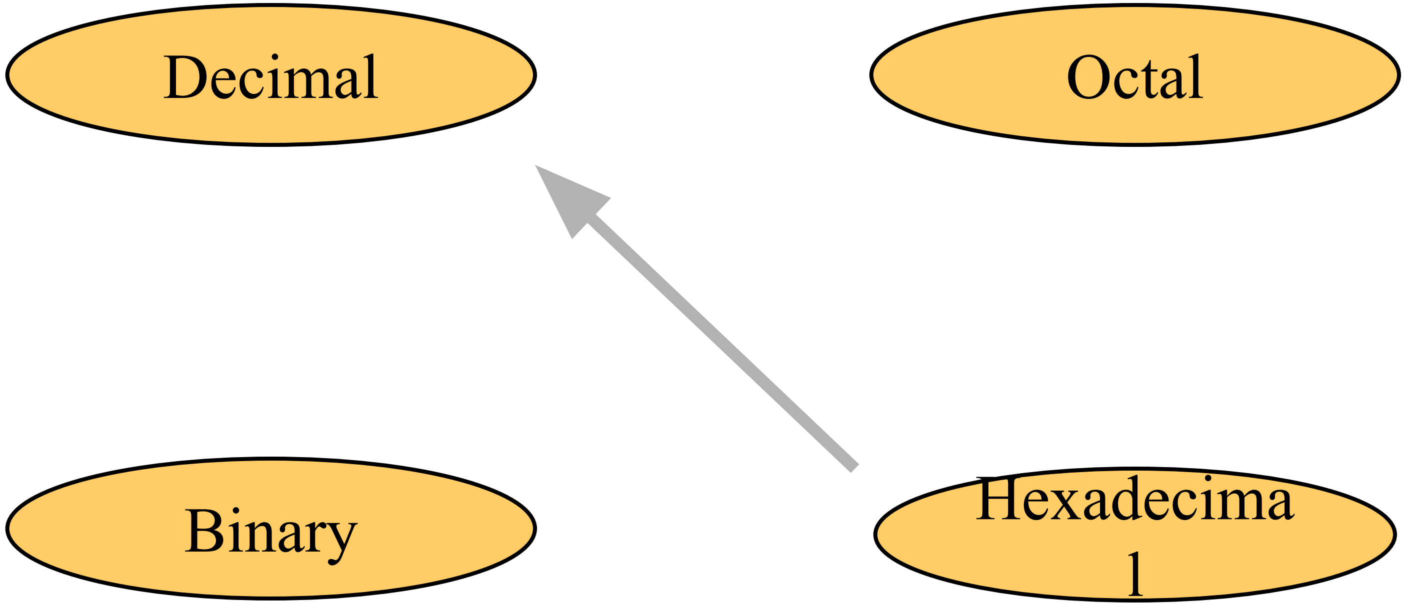
Example: Suppose 215_8 is an octal number, then it's decimal form will be,

- $215_8 = 2 \times 8^2 + 1 \times 8^1 + 5 \times 8^0$
- $= 2 \times 64 + 1 \times 8 + 5 \times 1 = 128 + 8 + 5$
- $= 141_{10}$

Example: Let 125 is an octal number denoted by 125_8. Find the decimal number.

- $125_8 = 1 \times 8^2 + 2 \times 8^1 + 5 \times 8^0$
- $= 1 \times 64 + 2 \times 8 + 5 \times 1 = 64+16+5$
- $=85_{10}$
-

Hexadecimal to Decimal



Hexadecimal to Decimal

- Technique
  - Multiply each bit by 16^n, where n is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results
  - Hexadecimal no – 0-9 & A-F

Example

$$
\begin{array}{rl}
ABC_{16} \Rightarrow C \times 16^0 = 12 \times 1 = & 12 \\
B \times 16^1 = 11 \times 16 = & 176 \\
A \times 16^2 = 10 \times 256 = & 2560 \\
\hline
& 2748_{10}
\end{array}
$$

Example

$\mathbf{(\mathit{A}10)_{16}=(\underline{\ \ })_{10}}$

$=A\times16^2+1\times16^1+0\times16^0$

$=10\times256+1\times16+0\times1$

$=2576$

Example

$(\boldsymbol{BCA})_{\mathbf{16}} = (\ )_{\mathbf{10}}$

$= B\times16^2 + C\times16^1 + A\times16^0$

$= 11\times256 + 12\times16 + 10\times1$

$= 3018$

Example

Example 1: What is 5C6 (Hexadecimal)?

- Solution: Step 1: The "5 " is the "16 x 16" position, so that means 5 x16 x16
- Step 2: The 'C' (12) is in the "16" position, so that means 12 x 16.
- Step 3: The "6" in the "1" position so that means 6.
- Answer is : 5C6 = 5 x 16 x 16 + 12 x 16 +6 = (1478) in Decimal.

Example 2: What is 3C5 (Hexadecimal)?

- Solution: Step 1: The "3 " is the "16 x 16" position, so that means 3 x16 x16
- Step 2: The 'C' (12) is in the "16" position, so that means 12 x 16.
- Step 3: The "5" is in the "1" position so that means 5.
- Answer is : 3C5 = 3 x 16 x 16 + 12 x 16 + 5 = (965) in Decimal.

Example

Example 3: What is 7B5 (Hexadecimal)?

- Solution: Step 1: The "7 " is the "16 x 16" position, so that means 7 x16 x16
- Step 2: The 'B' (11) is in the "11" position, so that means 11 x 16.
- Step 3: The 5" in the "1" position so that means 5.
- Answer is : 7B5 = 7 x 16 x 16 + 11 x 16 +5 = (1973) in Decimal.

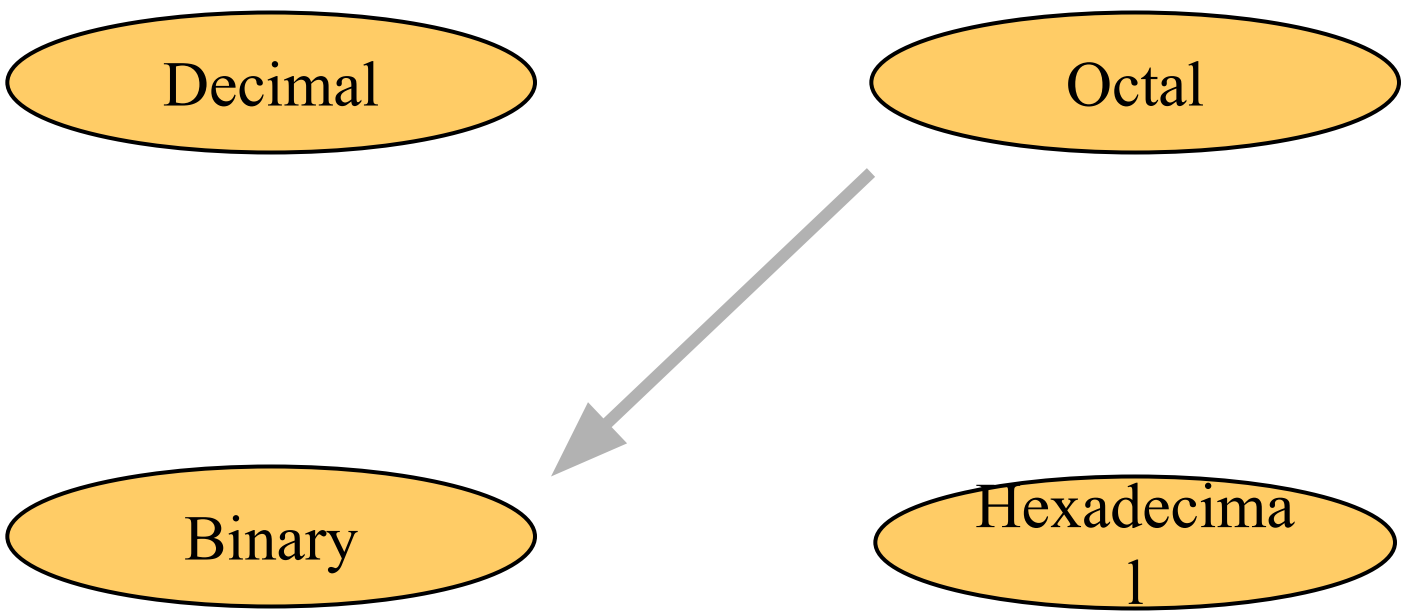
Example 4: What is 2E8 (Hexadecimal)?

- Solution: Step 1: The "2 " is the "16 x 16" position, so that means 2 x16 x16
- Step 2: The 'E' (14) is in the "16" position, so that means 14 x 16.
- Step 3: The "2" is in the "1" position so that means 2.
- Answer is : 2E8 = 2 x 16 x 16 + 14 x 16 +8 = (744) in Decimal.

Example 5: What is 4F8 (Hexadecimal)?

- Solution: Step 1: The "4 " is the "16 x 16" position, so that means 4 x16 x16
- Step 2: The 'F' (15) is in the "16" position, so that means 15 x 16.
- Step 3: The "8" is in the "1" position, which means 8.
- Answer is : 4F8 = 4 x16 x 16 + 15 x 16 +8 = (1272) in Decimal.

Octal to Binary



Octal to Binary

- Technique
  - Convert each octal digit to a 3-bit equivalent binary representation

Example

$705_8 = ?_2$

$$\begin{array}{ccc} 7 & 0 & 5 \\ \downarrow & \downarrow & \downarrow \\ 111 & 000 & 101 \end{array}$$

$705_8 = 111000101_2$

Example

$\mathbf{(123)_8 = (\underline{\quad ? \quad})_2}$

$$\begin{array}{ccc} 1 & 2 & 3 \\ \downarrow & \downarrow & \downarrow \\ 001 & 010 & 011 \end{array}$$

$\therefore (123)_8 = (1010011)_2$

Example

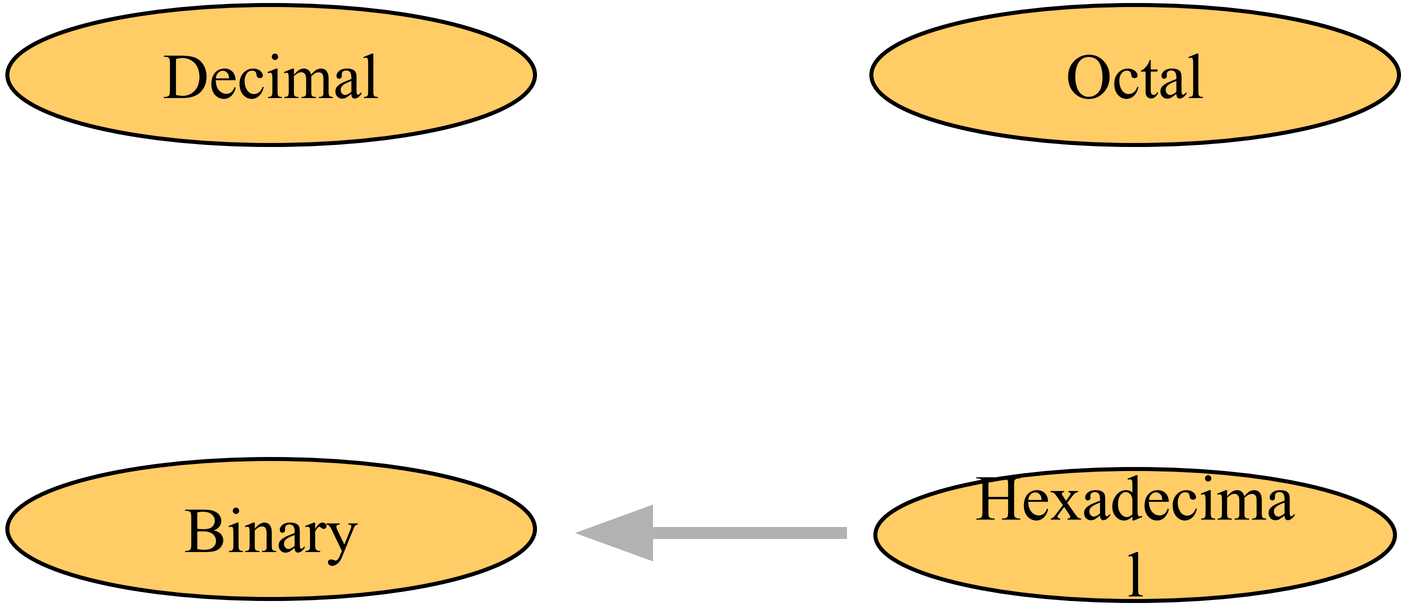
$(4321)_8 = (\underline{\quad ? \quad})_2$

4	3	2	1
↓	↓	↓	↓
100	011	010	001

$\therefore$

$(4321)_8 = (100011010001)_2$

Hexadecimal to Binary



Hexadecimal to Binary

- Technique
  - Convert each hexadecimal digit to a 4-bit equivalent binary representation

Example

$10AF_{16} = ?_2$

$$\begin{array}{cccc} 1 & 0 & A & F \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 0001 & 0000 & 1010 & 1111 \end{array}$$

$10AF_{16} = 0001000010101111_2$

Example

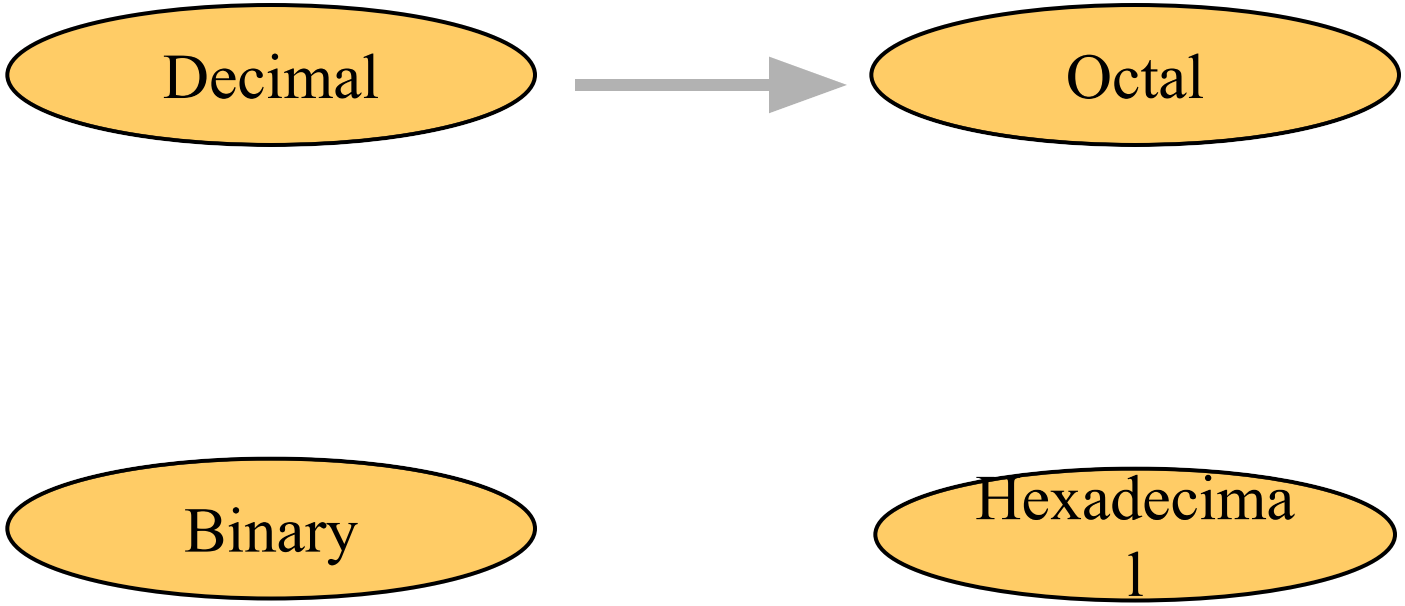
$\mathbf{(283)_{16} = (\underline{\quad\quad ? \quad\quad})_2}$

2	8	3
<u>0010</u>	<u>1000</u>	<u>0011</u>

$\therefore$

$(283)_{16}=(1010000011)_2$

Decimal to Octal



Decimal to Octal

- Technique
  - Divide by 8
  - Keep track of the remainder

Example

$1234_{10} = ?_8$

$$
\begin{array}{r|l r}
8 & 1234 & \\
\hline
8 & 154 & 2 \\
\hline
8 & 19 & 2 \\
\hline
8 & 2 & 3 \\
\hline
 & 0 & 2
\end{array}
$$

$1234_{10} = 2322_8$

Example

$(425)_{10} = (\underline{\quad ? \quad})_8$

8	425		
8	53	1	↑
8	6	5	↑
	0	6	↑

$\therefore$

$(425)_{10}=(651)_8$

Example

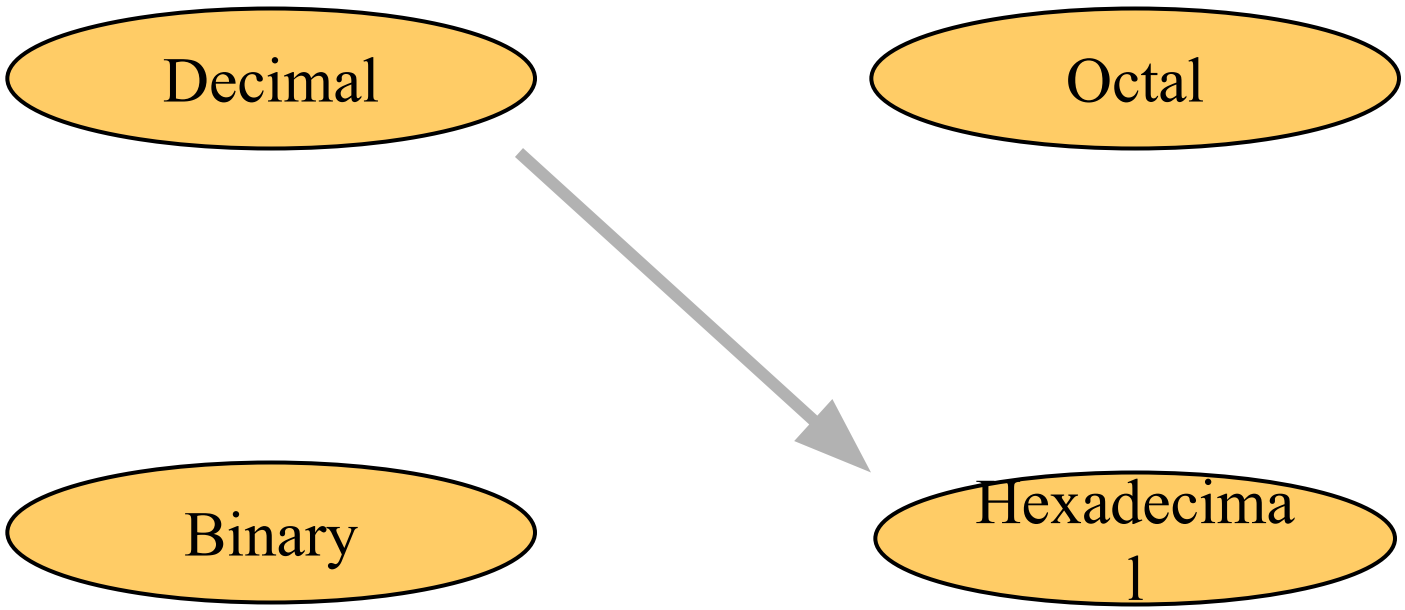
$(6260)_{10} = (\underline{\quad ? \quad})_8$

8	6260		
8	782	4	↑
8	97	6	↑
8	12	1	↑
8	1	4	↑
	0	1	↑

$\therefore$

$(6260)_{10} = (14164)_8$

Decimal to Hexadecimal



Decimal to Hexadecimal

- Technique
  - Divide by 16
  - Keep track of the remainder

Example

$1234_{10} = ?_{16}$

$$
\begin{array}{r|ll}
16 & 1234 & \\
\hline
16 & 77 & 2 \\
\hline
16 & 4 & 13 = D \\
\hline
 & 0 & 4
\end{array}
$$

$1234_{10} = 4D2_{16}$

Example

$\mathbf{(1423)_{10} = (\underline{\quad ? \quad})_{16}}$

16	1423		
16	88	F	↑
16	5	8	↑
	0	5	↑

$$\therefore (1423)_{10} = (58F)_{16}$$

Example

$(93419)_{10} = (\underline{\quad ? \quad})_{16}$

16	93419		
16	5838	B	↑
16	364	E	↑
16	22	C	↑
16	1	6	↑
	0	1	↑

$\therefore (93419)10=(16CEB)16$

Example

Convert 5386 to a hexadecimal number.

Number (Division)	Quotient	Remainder
5386 / 16	336	10 = A
336 / 16	21	0
21 / 16	1	5
1 / 16	0	1

Decimal Value $\longrightarrow$ Hexadecimal Value

$(5386)_{10} \longrightarrow (150\,A)_{16}$

Example

Convert $(960)_{10}$ into hexadecimal.

$$(960)_{10} = (3C0)_{16}$$

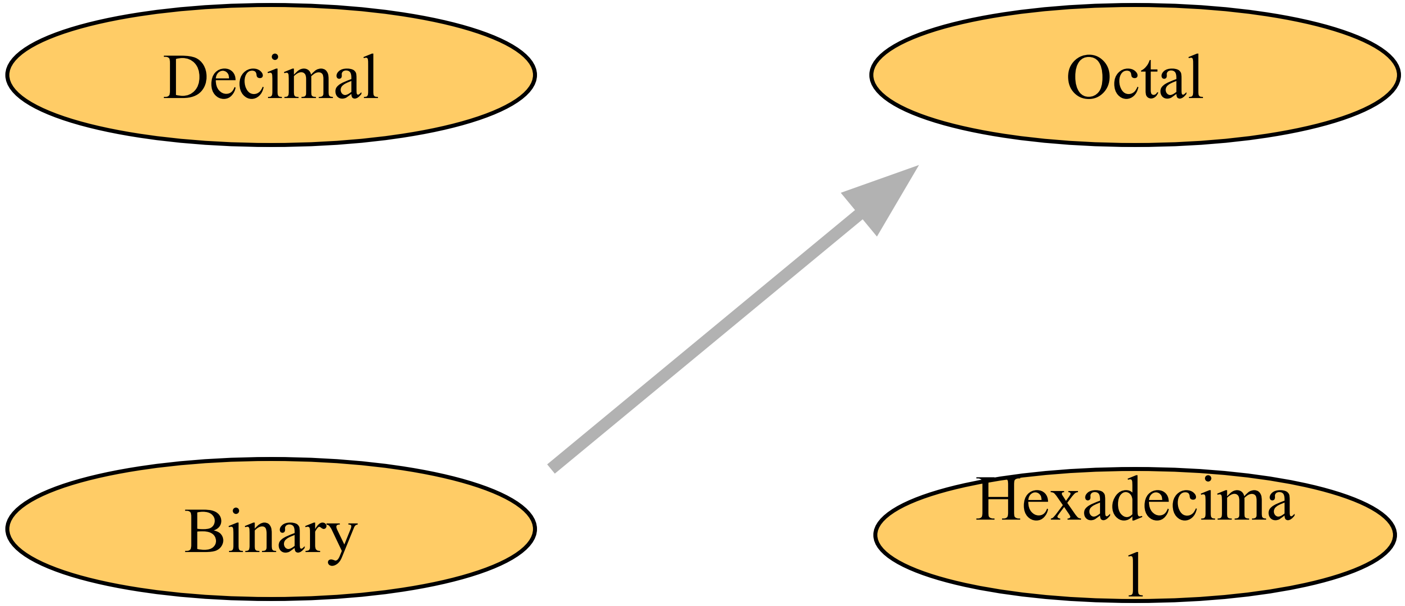
Convert 1228_{10} into hexadecimal.

$$1228_{10} = 4CC_{16}$$

Convert 600_{10} into a hexadecimal number.

$$600_{10} = 258_{16}$$

Binary to Octal



Binary to Octal

- Technique
  - Group bits in threes, starting on right
  - Convert to octal digits

Example

$1011010111_2 = ?_8$

1 011 010 111

1 3 2 7

$1011010111_2 = 1327_8$

Example

$\mathbf{(11011001)_2 = (\underline{\quad ? \quad})_8}$

$\underline{011}$	$\underline{011}$	$\underline{001}$
3	3	1

$$\therefore (11011001)_2 = (331)_8$$

Example

$\mathbf{(10110011)_2 = (\underline{\quad\quad ? \quad\quad})_8}$

$\underline{010}$	$\underline{110}$	$\underline{011}$
2	6	3

$$\therefore (10110011)_2 = (263)_8$$

Example

Example: Convert binary number 1010111100 into octal number.

Therefore, Binary to octal is.

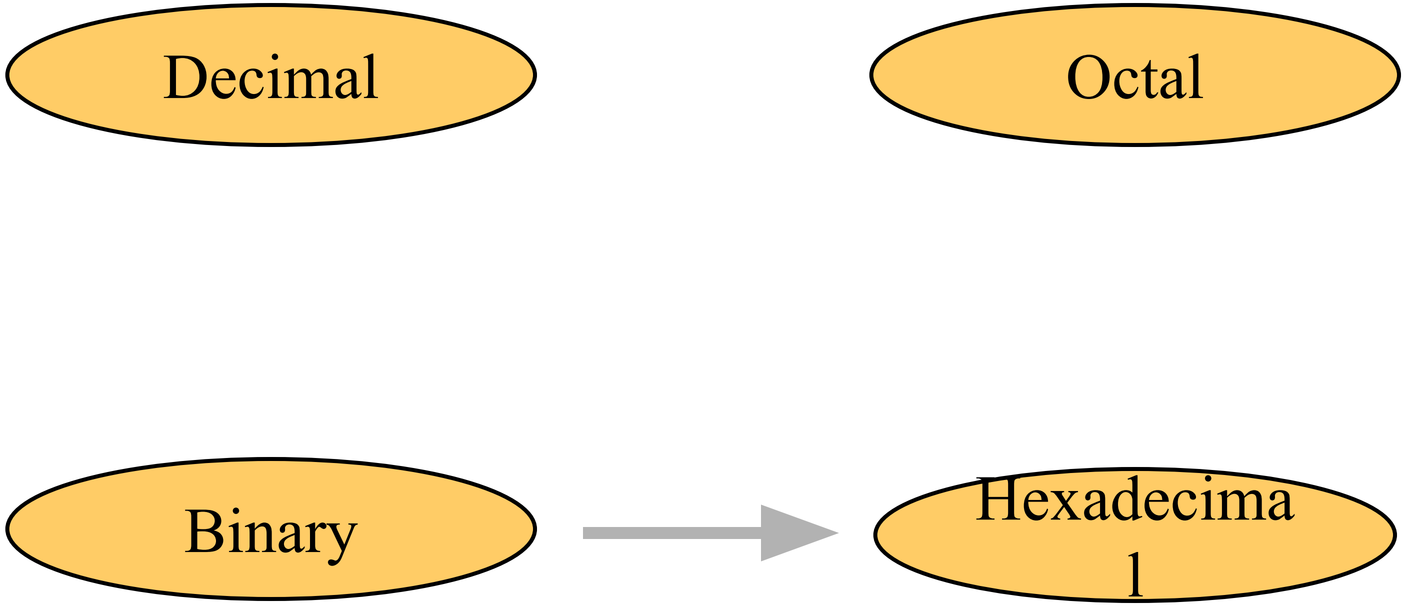
= (1010111100)

= (001 010 111 100)

= (1 2 7 4)

= (1274)

Binary to Hexadecimal



Binary to Hexadecimal

- Technique
  - Group bits in fours, starting on right
  - Convert to hexadecimal digits

Example

$1010111011_2 = ?_{16}$

10 1011 1011

2 B B

$1010111011_2 = 2BB_{16}$

Example

$$\mathbf{(1111010100)_2 = (\underline{\quad ? \quad})_{16}}$$

$\underline{0011}$	$\underline{1101}$	$\underline{0100}$
3	D	4

$$\therefore (1111010100)_2 = (3D4)_{16}$$

Example

2. $(111101010011110)_2 = (\underline{\quad ? \quad})_{16}$

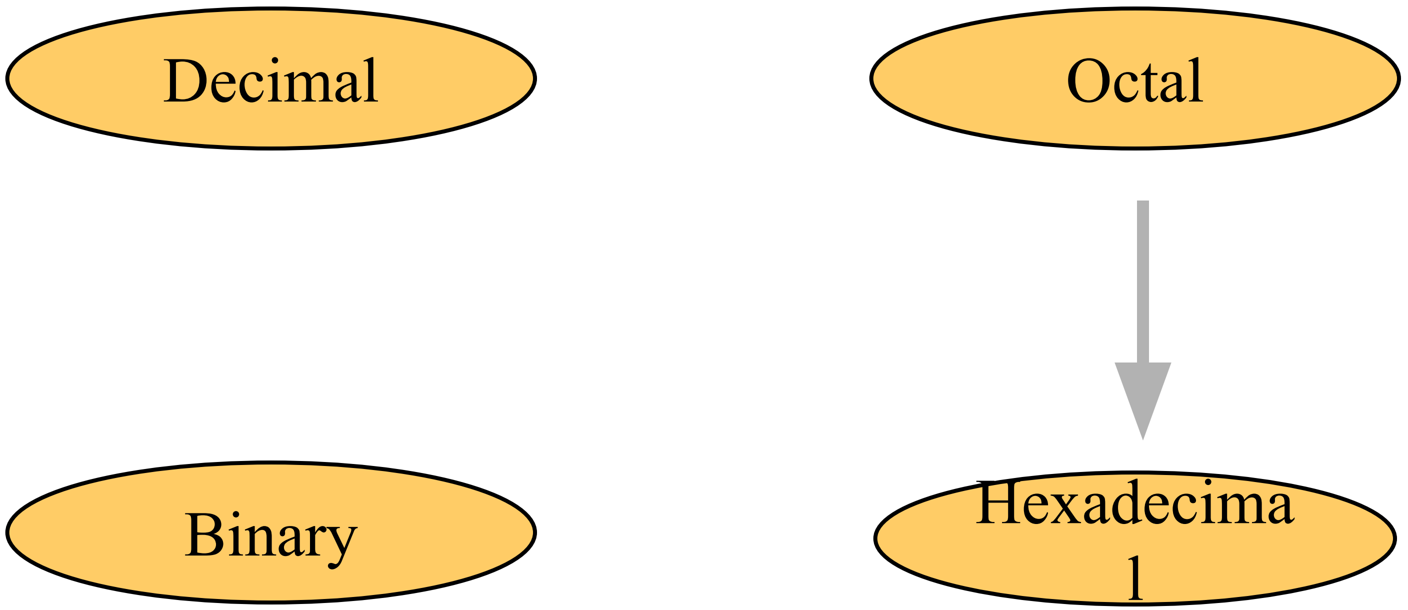
Solution:

$$(111101010011110)_2 = \left(\quad\right)_{16}$$

$$\underbrace{\underline{1111}}_{F}\ \underbrace{\underline{0101}}_{5}\ \underbrace{\underline{0011}}_{3}\ \underbrace{\underline{1110}}_{E}$$

$$\therefore (111101010011110)_2 = \left(F53E\right)_{16}$$

Octal to Hexadecimal



Octal to Hexadecimal

- Technique
  - Use binary as an intermediary
  - Octal -> binary then binary -> hexadecimal

Example

$1076_8 = ?_{16}$

1 0 7 6

001 000 111 110

2 3 E

$1076_8 = 23E_{16}$

Example

$(567)_8 = (\underline{\quad ? \quad})_{16}$

First convert octal to binary

$(567)_8=(\underline{\ })_2$

5	6	7
101	110	111

$\therefore$

$(567)_8=(101110111)_2$

Now convert binary to hexadecimal

$(101110111)_2=(\underline{\ })_1$

0001	0111	0111
1	7	7

$\therefore$

$(101110111)_2=(177)_{16}$

$\therefore$

$(567)_8=(177)_{16}$

Example

$$(4321)_8 = (\underline{\quad ? \quad})_{16}$$

First convert octal to binary

$$(4321)_8 = ()_2$$

4	3	2	1
100	011	010	001

$\therefore (4321)_8 = (100011010001)_2$

Now convert binary to hexadecimal

$$(100011010001)_2 = ()_{16}$$

1000	1101	0001
8	D	1

$\therefore$

$(100011010001)_2 = (8D1)_{16}$

$\therefore$

$(4321)_8 = (8D1)_{16}$

Example

Convert 536 from octal to hexadecimal number

Convert 536(octal) into its binary equivalent we get

$(536)_8$ = (101) (011) (110)

$=(101011110)_2$

Now forming the group of 4 binary bits to obtain its hexadecimal equivalent,

$(101011110)_2$= (0001) (0101) (1110)

= $(15E)_{16}$

So the hexadecimal number of 536 is 15E.

Example

Convert 752 from octal to hexadecimal number

Step 1:

Octal to Binary Conversion

7	5	2
111	101	010

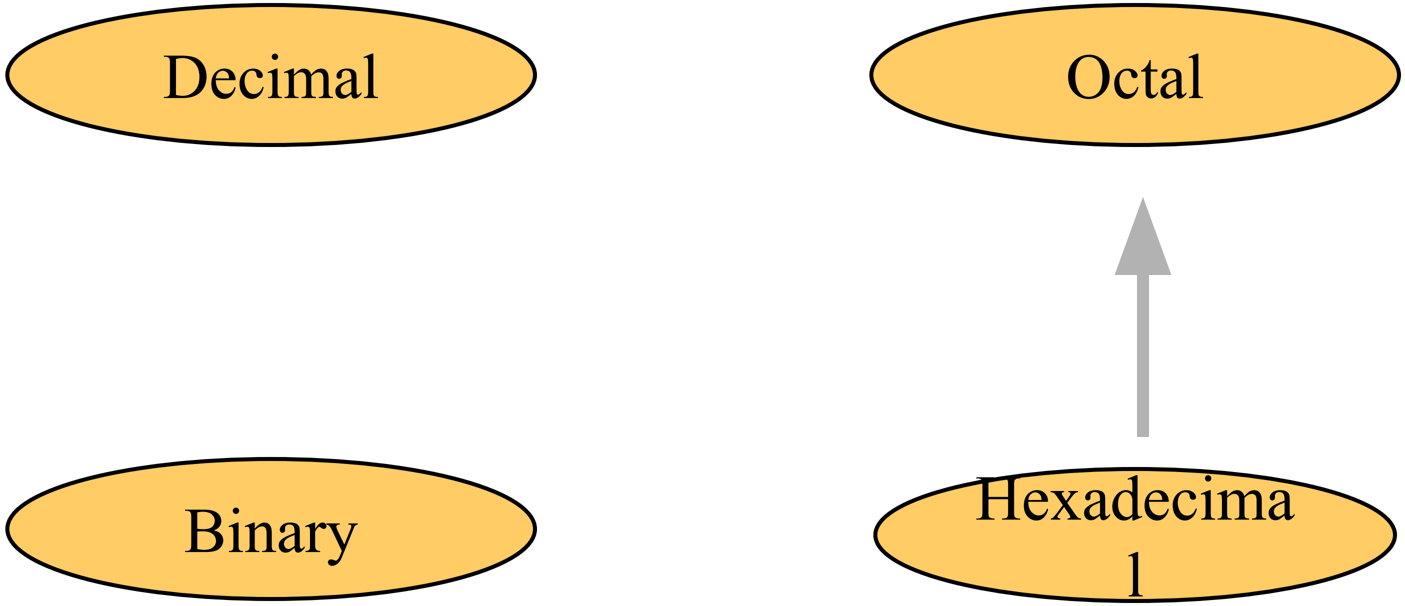
So the binary equivalent is 111101010

Step 2:

Binary to Hex Conversion

<u>0001</u>	<u>1110</u>	<u>1010</u>
1	D	9

Hexadecimal to Octal



Hexadecimal to Octal

- Technique
  - Use binary as an intermediary

Example

$1F0C_{16} = ?_{8}$

1	F	0	C
0001	1111	0000	1100

0 | 001 | 111 | 100 | 001 | 100

1 | 7 | 4 | 1 | 4

$1F0C_{16} = 17414_{8}$

Example

Example: Convert $1BC_{16}$ into an octal number.

Solution: Given, $1BC_{16}$ is a hexadecimal number.

1 → 0001, B → 1011, C →1100

Now group them from right to left, each having 3 digits.

000, 110, 111, 100

000→0, 110 →6, 111→7, 100→4

Hence, $1BC_{16} = 674_8$

Example

$(951)_{16}$ = (____?____)$_8$

First convert hexadecimal to binary

$(951)_{16}=()_2$

9	5	1
$\underline{1001}$	$\underline{0101}$	$\underline{0001}$

$\therefore$

$(951)_{16}=(100101010001)_2$

Now convert binary to octal

$(100101010001)_2=()_8$

$\underline{100}$	$\underline{101}$	$\underline{010}$	$\underline{001}$
4	5	2	1

$\therefore (100101010001)_2=(4521)_8$

$\therefore$

$(951)_{16}=(4521)_8$

Example

$(FC3A)_{16}$ = (____?____)$_8$

First convert hexadecimal to binary

$(FC3A)_{16} = ()_2$

F	C	3	A
1111	1100	0011	1010

$\therefore (FC3A)_{16} = (1111110000111010)_2$

Now convert binary to octal

$(1111110000111010)_2 = ()_8$

001	111	110	000	111	010
1	7	6	0	7	2

$\therefore$
$(1111110000111010)_2 = (176072)_8$

$\therefore$
$(FC3A)_{16} = (176072)_8$

Example

Convert the following hexadecimal number to octal number $2CD_{16}$.

Given, $2CD_{16}$ is a hexadecimal number.

$2 \rightarrow 0010$, $C \rightarrow 1100$, $D \rightarrow 1101$,

Now you will be grouping them from right to left, each having 3 digits.

001, 011, 001, 101

$001 \rightarrow 1$, $011 \rightarrow 3$, $001 \rightarrow 1$, $101 \rightarrow 5$

Hence, $2CD_{16} = 13158$

Example

Convert the following hexadecimal number to octal number $3EC_{16}$.

$3EC_{16}$ is a hexadecimal number.

$3 \rightarrow 0010$, $E \rightarrow 1110$, $C \rightarrow 1100$,

Now you will be grouping them from right to left, each having 3 digits.

001, 011, 101, 100

$001 \rightarrow 1$, $011 \rightarrow 3$, $101 \rightarrow 5$, $100 \rightarrow 4$

Hence, $3EC_{16} = 1354_8$

Example

Convert $ABCD_{16}$ to equivalent octal form.
Convert 912_{16} to equivalent octal form.

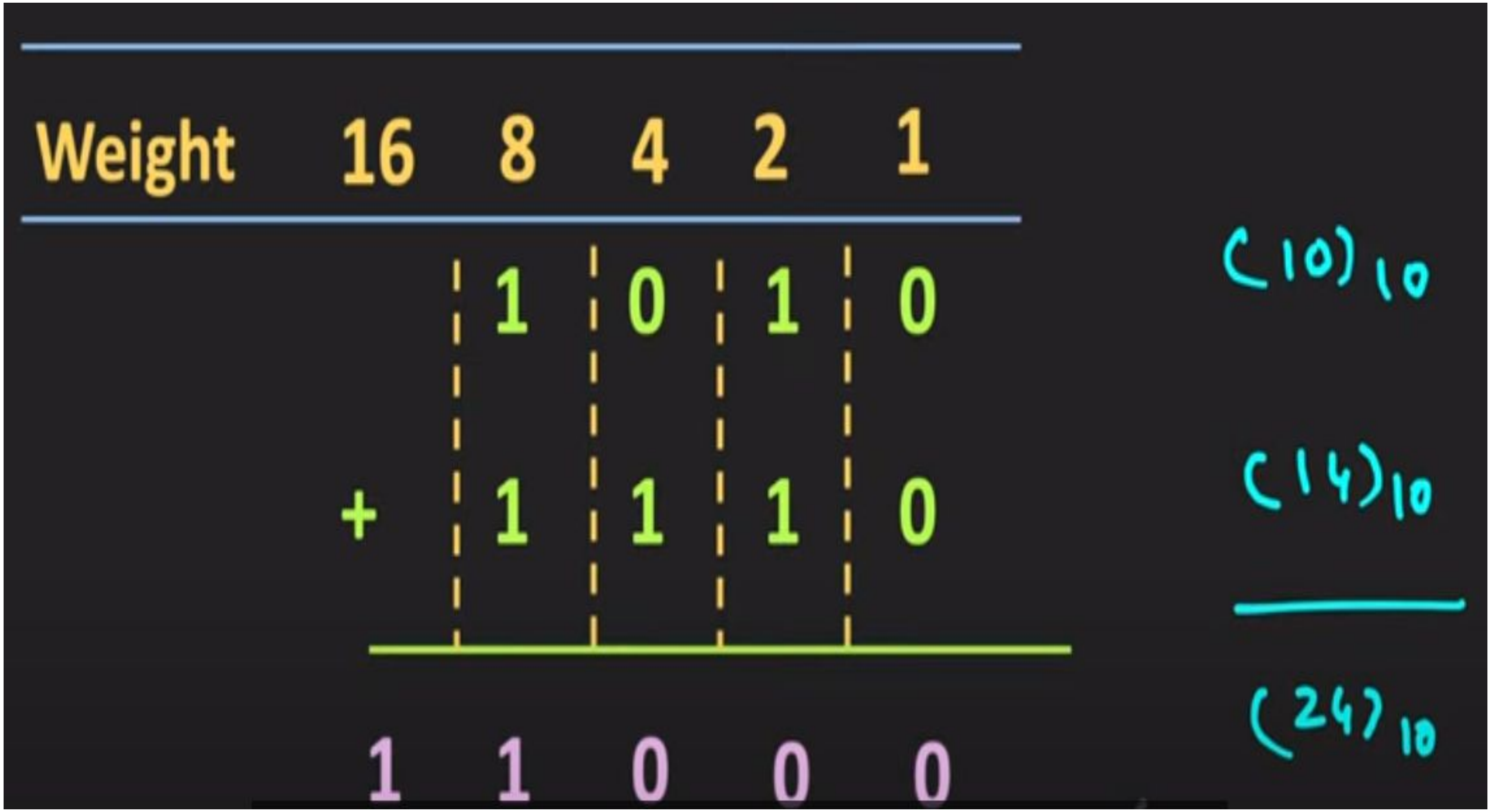
Binary Addition (1 of 2)

- **What is Binary Addition**
- The binary addition operation works similarly to the base 10 decimal system, except that it is a base 2 system.
- The binary system consists of only two digits, 1 and 0.
- Most of the functionalities of the computer system use the binary number system.
- The binary code uses the digits 1’s and 0’s to make certain processes turn off or on.
- The process of the addition operation is very familiar to the decimal system by adjusting to the base 2.
- Before attempting the binary addition process, we should have complete knowledge of how the place works in the binary number system.
- Because most of the modern digital computers and electronic circuits perform the binary operation by representing each bit as a voltage signal.
- The bit 0 represents the “OFF” state, and the bit 1 represents the “ON” state.

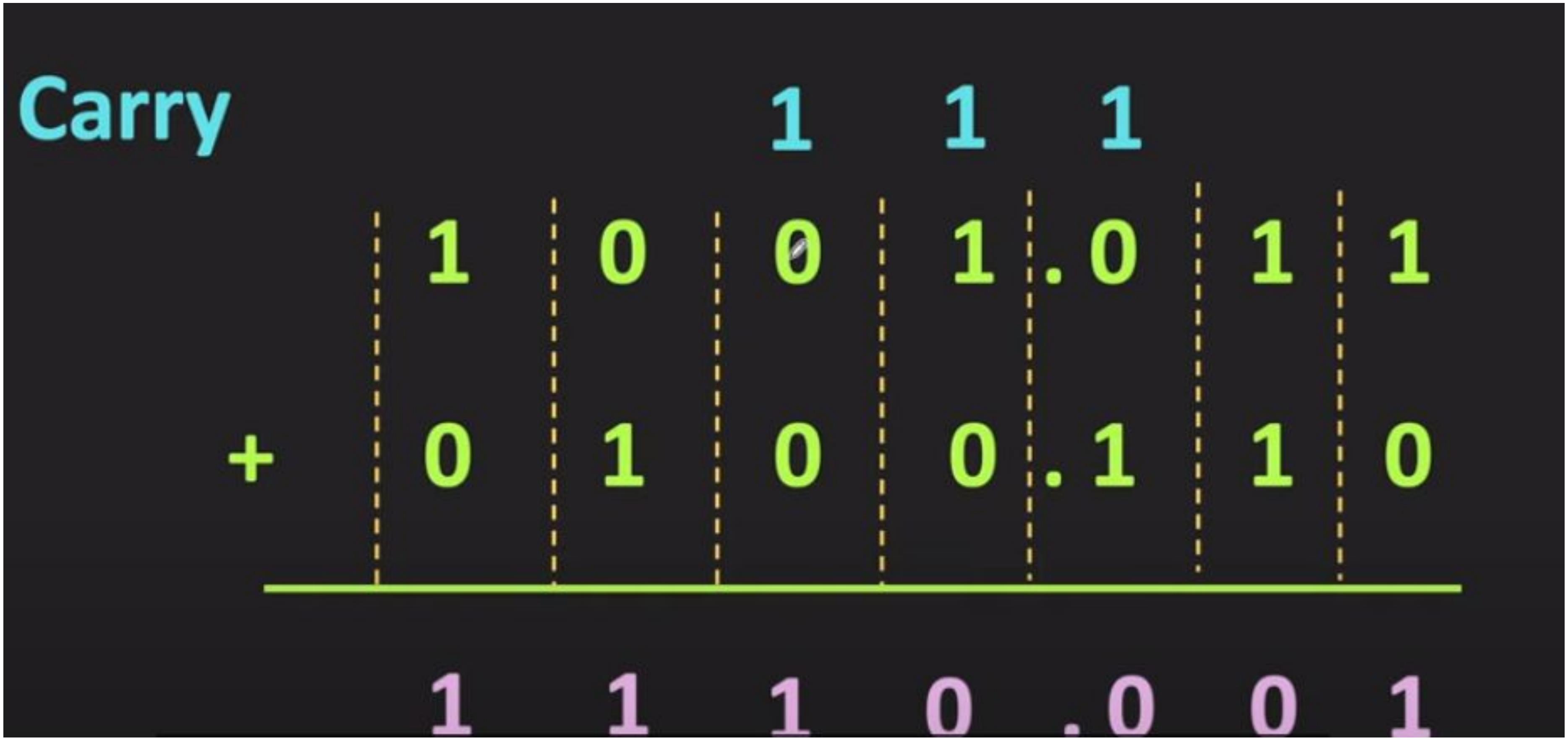
Rules of Binary Addition

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 =10 (carry 1 to the next significant bit)
- 1 + 1 + 1 = 11(carry 1 to the next significant bit)

Binary Addition



Binary Addition



Binary Addition

$101_2 + 10_2 =$

1 0 1

+ 1 0

………..

1 1 1

Binary Addition

$1001_2 + 111_2 =$

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

Binary Addition

$1010_2 + 1111_2 =$

```
      1  1
      1   0   1  0
+      1  1   1  1
-----------------
     1 1  0   0  1
```

Binary Addition

Add: 10011_2 and 110001_2

```
      1     1 1
        1 0 0 1 1
      +   1 1 0 0 0 1
-----------------------------
    1 0 0 0 1 0 0
```

Binary Addition

Add: 1 + 11 (100)
1010 + 11 (1101)
100101 + 10101 (111010)

Binary subtraction

- Binary subtraction is one of the four binary operations, where we perform the subtraction method for two binary numbers (comprising only two digits, 0 and 1).
- This operation is similar to the basic arithmetic subtraction performed on decimal numbers in Maths.
- Hence, when we subtract 1 from 0, we need to borrow 1 from the next higher order digit, to reduce the digit by 1 and the remainder left here is also 1.

Binary Subtraction Rules

Binary Number	**Subtraction Value**
0 – 0	0
1 – 0	1
0 – 1	1 (Borrow 1 from the next high order digit)
1 – 1	0

Binary Subtraction Examples

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

Binary Subtraction Examples

Example : 1010 – 101

1 1 borrow

1 0 1 0
\- 1 0 1
——————
0 1 0 1

Binary Subtraction Examples

Example : 1001 – 101

1 borrow

1 0 0 1
- 1 0 1
——————
0 1 0 0

Binary Subtraction Examples

Example : Subtract 1101_2 from 10110_2.

1 1 borrow

1 0 1 1 0
- 1 1 0 1
——————
0 1 0 0 1

Binary Addition

$1011011 - 10010 = 1001001$
$1010110 - 101010 = 101100$
$101101 - 100111 = 110$
$100010110 - 1111010 = 10011100$
$1000101 - 101100 = 11001$
$1110110 - 1010111 = 11111$

Number System

To a computer, everything is a number, i.e., alphabets, pictures, sounds, etc., are numbers. Number system is categorized into four types −

- Binary number system consists of only two values, either 0 or 1
- Octal number system represents values in 8 digits.
- Decimal number system represents values in 10 digits.
- Hexadecimal number system represents values in 16 digits.

System	Base	Digits
Binary	2	0 1
Octal	8	0 1 2 3 4 5 6 7
Decimal	10	0 1 2 3 4 5 6 7 8 9
Hexadecimal	16	0 1 2 3 4 5 6 7 8 9 A B C D E F

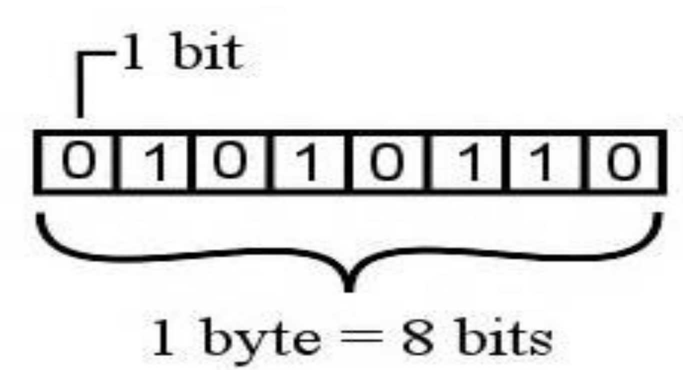
Number System

- The smallest unit of information that can be stored by a computer is a *binary digit* (usually shortened to "bit").
- Its value is usually held in memory as an electrical charge stored in a capacitor. Modern memory chips contain millions of these tiny capacitors, each of which is capable of storing exactly one bit of information.
- A single bit can have one of two values at any given time - *one* or *zero*.
- As we shall see, in order to represent a number greater than one, we will have to use several bits.

Bits and Bytes

Bits – A bit is a smallest possible unit of data that a computer can recognize or use. Computer usually uses bits in groups.

Bytes – group of eight bits is called a byte. Half a byte is called a nibble.



Bits and Bytes

- Generally speaking, single bits are used only for storing *Boolean values* (true or false).
- In most programming languages, *true* equates to *one*, while *false* equates to *zero*.
- The definition of the byte has varied over the years but it is now generally considered to be a group of eight bits, and can be used to represent alpha-numeric and non-printing characters, unsigned integer (whole number) values from 0 to 255, or signed integer values from -127 to +127.

Bits and Bytes

- The number of bits that can be processed by a CPU in a single machine operation is dependent upon the number of bits it can store in its internal registers.
- In the early days of computing this was a relatively small number (four or eight bits).
- At some point, therefore, the size of the processor's register coincided with the size of a byte.
- For many years now, however, this has not been the case.
- As CPU architecture has advanced, we have seen the size of registers double and re-double. Most processors now have either 32-bit or 64-bit registers that can hold four or eight bytes of data respectively.

How Many Bits Are Necessary to Represent Something?

- 1 bit can represent two (2^1) symbols
  - either a 0 or a 1
- 2 bits can represent four (2^2) symbols
  - 00 or 01 or 10 or 11
- 3 bits can represent eight (2^3) symbols
  - 000 or 001 or 011 or 111 or 100 or 110 or 101 or 010
- 4 bits can represent sixteen (2^4) symbols
- 5 bits can represent 32 (2^5) symbols
- 6 bits can represent 64 (2^6) symbols
- 7 bits can represent 128 (2^7) symbols
- 8 bits (a byte) can represent 256 (2^8) symbols
- n bits can represent (2^n) symbols!

Bits and Bytes

A modern CPU has a 64-bit architecture

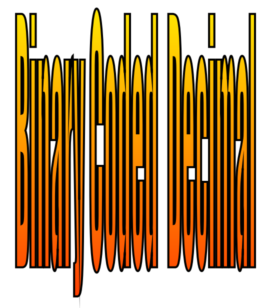
The following table shows conversion of Bits and Bytes –

Byte Value	Bit Value
1 Byte	8 Bits
1024 Bytes	1 Kilobyte
1024 Kilobytes	1 Megabyte
1024 Megabytes	1 Gigabyte
1024 Gigabytes	1 Terabyte
1024 Terabytes	1 Petabyte
1024 Petabytes	1 Exabyte
1024 Exabytes	1 Zettabyte
1024 Zettabytes	1 Yottabyte
1024 Yottabytes	1 Brontobyte
1024 Brontobytes	1 Geopbytes

Binary Coded Decimal (BCD)

- *Representing* one decimal number at a time
- How can we *represent* the ten decimal numbers (0-9) in binary code?

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



- We can represent any integer by a string of binary digits.
- For example, 749 can be *represented* in binary as: 011101001001

Binary Coded Decimal (BCD)

- **Binary Coded Decimal**, or **BCD**, is another process for converting decimal numbers into their binary equivalents.
- It is noticeable that the BCD is nothing more than a binary representation of each digit of a decimal number.

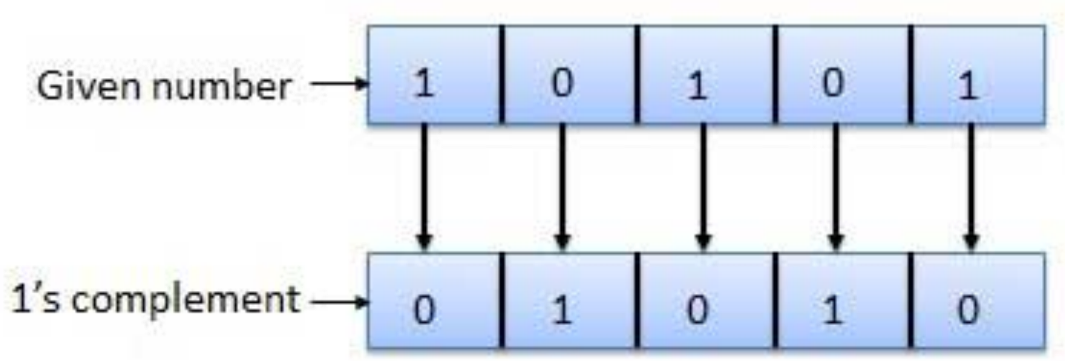
Complements

- Complements are used in the digital computers in order to simplify the subtraction operation and for the logical manipulations.
- For each radix-r system (radix r represents base of number system) there are two types of complements.

S.N.	Complement	Description
1	Radix Complement	The radix complement is referred to as the r's complement
2	Diminished Radix Complement	The diminished radix complement is referred to as the (r-1)'s complement

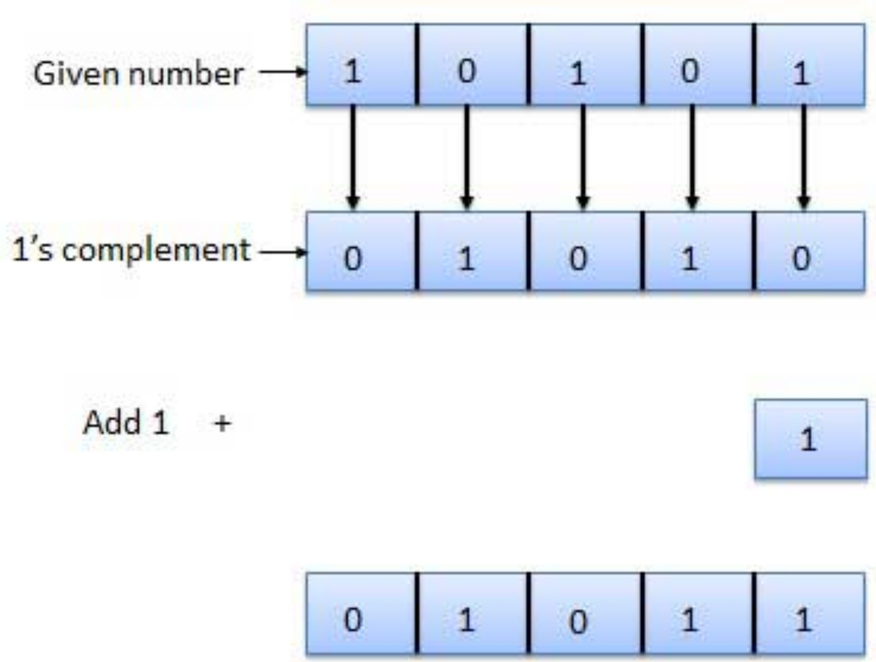
Binary system complements

- As the binary system has base r = 2. So the two types of complements for the binary system are 2's complement and 1's complement.
- **1's complement**
- The 1's complement of a number is found by changing all 1's to 0's and all 0's to 1's. This is called as taking complement or 1's complement.
- Example of 1's Complement is as follows.



Binary system complements

- **2's complement**
- The 2's complement of binary number is obtained by adding 1 to the Least Significant Bit (LSB) of 1's complement of the number.
- 2's complement = 1's complement + 1
- Example of 2's Complement is as follows.



Binary subtraction using 1’s complement

The 1's complement of a number is obtained by interchanging every 0 to 1 and every 1 to 0 in a binary number.

For example, the 1's complement of the binary number 110 is 001.

To perform binary subtraction using 1's complement, please follow the below steps

1. At first, find 1's complement of the B(subtrahend).
2. Then add it to the A(minuend).
3. If the final carry over of the sum is 1, then it is dropped and 1 is added to the result.
4. If there is no carry over, then 1's complement of the sum is the final result and it is negative.

Complements are used in digital circuits, because it is faster to subtract by adding complements than by performing true subtraction.

Binary subtraction using 1’s complement

- Subtract 110010 -100101
  - Step 1: Find out the 1's complement of the subtrahend (37), which is **011010**
  - Step 2: Add it with the minuend(50), which is 110010

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

  - The left-most digit 1 is a carryover of this addition. Since there is a carryover we
    it with the result, which is 00110

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

Binary subtraction using 1's complement

Find Subtraction of 110 and 101 using 1's complement method

–Step 1: Find out the 1's complement of the subtrahend which is **010**

–Step 2: Add it with the minuend which is 110

```
   1
   1 1 0
 + 0 1 0
 ________
  1 0 0 0
```

–The left-most digit 1 is a carryover of this addition. Since there is a carryover we add
with the result, which is 000

```
   0 0 0
 +     1
 ________
   0 0 1
```

Binary subtraction using 1's complement

Find Subtraction of 10110 and 11101 using 1's complement method

Step 1: Find out the 1's complement of the subtrahend which is **00010**

Step 2: Add it with the minuend which is 10110

```
   1 1
 1 0 1 1 0
+0 0 0 1 0
__________
 1 1 0 0 0
```

–Here there is no carry, answer is - (1's complement of the sum obtained 11000)

–So answer is -00111

Binary subtraction using 1's complement

- **1011.001 – 110.10**
- **Solution:**
- 1's complement of 0110.100 is 1001.011 Hence
-

Minued -		1 0 1 1 . 0 0 1
1's complement of subtrahend -		1 0 0 1 . 0 1 1
Carry over -	1	0 1 0 0 . 1 0 0
		1
		0 1 0 0 . 1 0 1

- **Hence the required difference is 100.101**

Binary subtraction using 1's complement

- **10110.01 – 11010.10**
- **Solution:**
- 1's complement of 11010.10 is 00101.01
-

$$\begin{array}{r} 1\,0\,1\,1\,0\,.\,0\,1 \\ \underline{0\,0\,1\,0\,1\,.\,0\,1} \\ 1\,1\,0\,1\,1\,.\,1\,0 \end{array}$$

- **Hence the required difference is – 00100.01 i.e. – 100.01**

Binary subtraction using 2’s complement

Method : 2's complement subtraction steps :

1. At first, find 2's complement of the B(subtrahend).
2. Then add it to the A(minuend).
3. If the final carry over of the sum is 1, then it is dropped and the result is positive.
4. If there is no carry over, then 2's complement of the sum is the final result and it is negative.

Binary subtraction using 2’s complement

Find Subtraction of 110 and 101 using 2's complement method

Step 1 : 2's complement of a number is 1 added to it's 1's complement number.

101 -> 010 (1’s complement)

010

\+ 1

___

0 1 1 (2’s complement)

Step 2: Now Add this 2's complement to 110

  1

  1 1 0

\+ 0 1 1

____

 1 0 0 1

Step 3: The left most bit of the result is called carry and it is ignored.
So answer is 001

Binary subtraction using 2’s complement

Find Subtraction of 10110 and 11101 using 2's complement method

Step 1 : 2's complement of a number is 1 added to it's 1's complement number.

11101 -> 00010 (1’s complement)

```
00010
+    1
_______
00011 (2’s complement)
```

Step 2: Now Add this 2's complement to 10110

```
    11
  10110
+ 00011
________
  11001
```

Step 3: Here there is no carry, answer is - (2's complement of the sum obtained 11001)

00110 (1’s complement)

00110+1 =00111 (2’s complement)

So answer is -00111

Binary subtraction using 2’s complement

110110 - 10110

Solution:

- The numbers of bits in the subtrahend is 5 while that of minuend is 6. We make the number of bits in the subtrahend equal to that of minuend by taking a `0’ in the sixth place of the subtrahend.
- Now, 2’s complement of 010110 is (101001 + 1) i.e.101010. Adding this with the minuend.
-

	1 1 0 1 1 0	Minuend
	1 0 1 0 1 0	2’s complement of subtrahend
Carry over 1	1 0 0 0 0 0	Result of addition

- After dropping the carry over we get the result of subtraction to be 100000.

Binary subtraction using 2's complement

- **10110 – 11010**
- **Solution:**
- 2's complement of 11010 is (00101 + 1) i.e. 00110. Hence
- Minued - 1 0 1 1 0

  2's complement of subtrahend - <u>0 0 1 1 0</u>

  Result of addition - 1 1 1 0 0

- As there is no carry over, the result of subtraction is negative and is obtained by writing the 2's complement of 11100 i.e.(00011 + 1) or 00100.
- Hence the difference is – 100.

Binary subtraction using 2's complement

- **1010.11 – 1001.01**
- **Solution:**
- 2's complement of 1001.01 is 0110.11. Hence
- Minued - 1 0 1 0 . 1 1

2's complement of subtrahend - 0 1 1 0 . 1 1

Carry over 1 0 0 0 1 . 1 0

- After dropping the carry over we get the result of subtraction as 1.10.

Binary subtraction using 2's complement

- **10100.01 – 11011.10**
- **Solution:**
- 2's complement of 11011.10 is 00100.10. Hence
- Minued - 1 0 1 0 0 . 0 1

2's complement of subtrahend - 0 0 1 0 0 . 1 0

Result of addition - 1 1 0 0 0 . 1 1

- As there is no carry over the result of subtraction is negative and is obtained by writing the 2's complement of 11000.11.
- Hence the required result is – 00111.01.

Binary subtraction using 1's & 2's complement

- $(45)_{10} - (56)_{10} = (?)_2$ (with 2's complement)
- $(56)_{10} - (45)_{10} = (?)_2$ (with 1's complement)
- $(4545)_{10} - (5656)_{10} = (?)_2$ (with 2's complement)
- $(5656)_{10} - (4545)_{10} = (?)_2$ (with 1's complement)
- $(45.25)_{10} - (56.20)_{10} = (?)_2$ (with 2's complement)
- $(35.21)_{10} - (21.35)_{10} = (?)_2$ (with 1's complement)