Number Systems

Numbers

- Each number system is associated with a base or radix
- A number in *base r* contains r digits 0,1,2,...,r-1
  - Decimal (Base 10): 0,1,2,3,4,5,6,7,8,9
- Numbers are usually expressed in positional notation

Significant Digits

Binary: 11101101

Most significant digit ***Least significant digit***

Hexadecimal: 1D63A7A

Most significant digit ***Least significant digit***

Number Systems

System	Base	Symbols
Decimal	10	0, 1, … 9
Binary	2	0, 1
Octal	8	0, 1, … 7
Hexa-decimal	16	0, 1, … 9, A, B, … F

Quantities

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

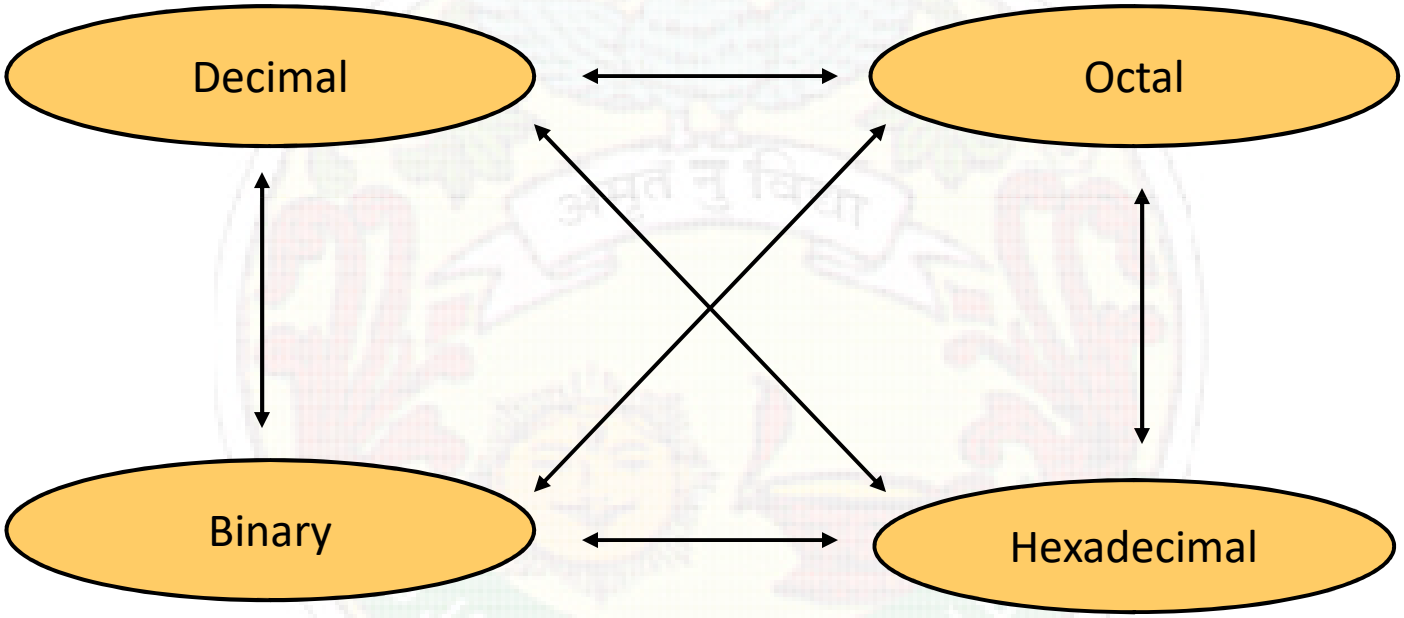
Decimal	Binary	Octal	Hexa-decimal
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Quantities

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

Conversion Among Bases

- The possibilities:

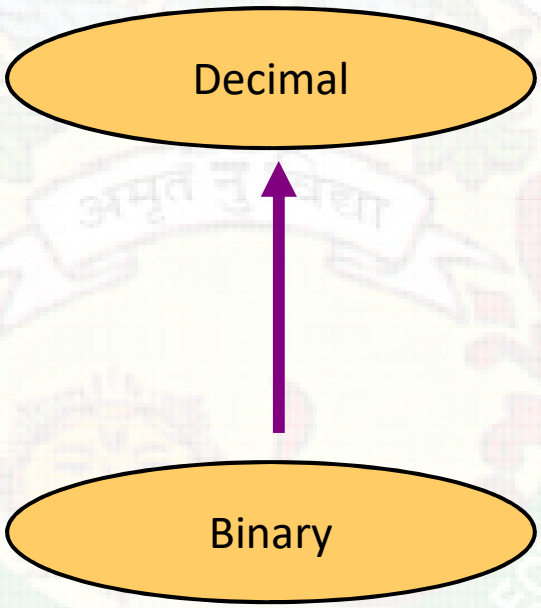


pp. 40-46

Example

$$25_{10} = 11001_2 = 31_8 = 19_{16}$$

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

101011_2 =>

$1 \times 2^0 = 1$

$1 \times 2^1 = 2$

$0 \times 2^2 = 0$

$1 \times 2^3 = 8$

$0 \times 2^4 = 0$

$1 \times 2^5 = 32$

43_{10}

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

724_8 =>

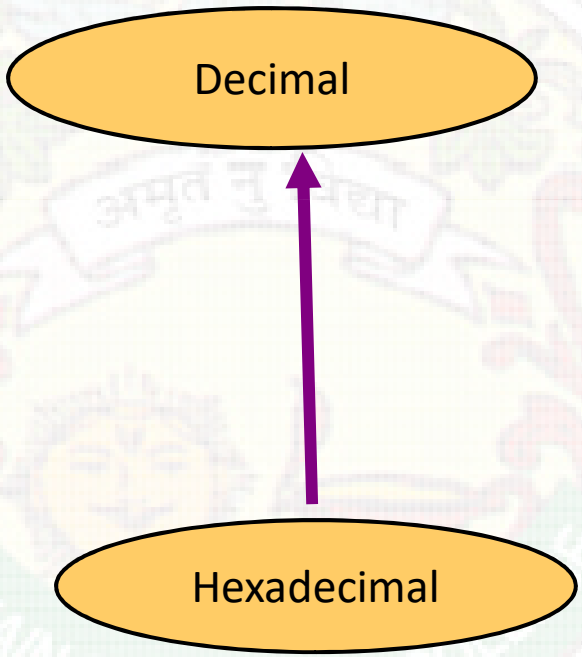
$4 \times 8^0 = 4$

$2 \times 8^1 = 16$

$7 \times 8^2 = 448$

468_{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

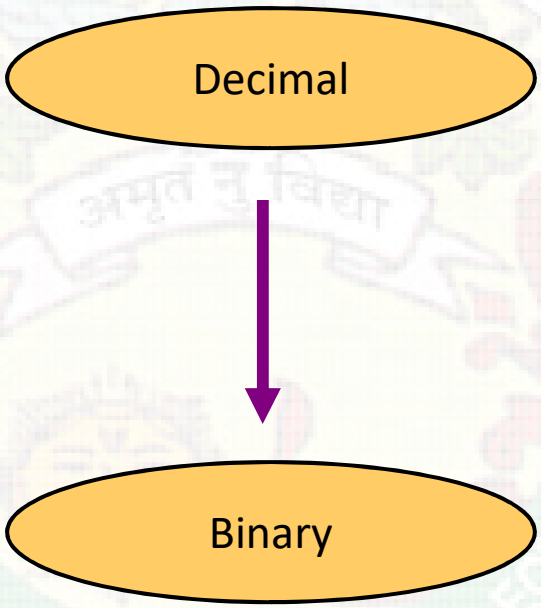
Example

ABC_{16} =>

C x 16^0 = 12 x 1 = 12
B x 16^1 = 11 x 16 = 176
A x 16^2 = 10 x 256 = 2560

2748_{10}

Decimal to Binary



Decimal to Binary

- Technique
  - Divide by two, keep track of the remainder
  - First remainder is bit 0 (LSB, least-significant bit)
  - Second remainder is bit 1
  - Etc.

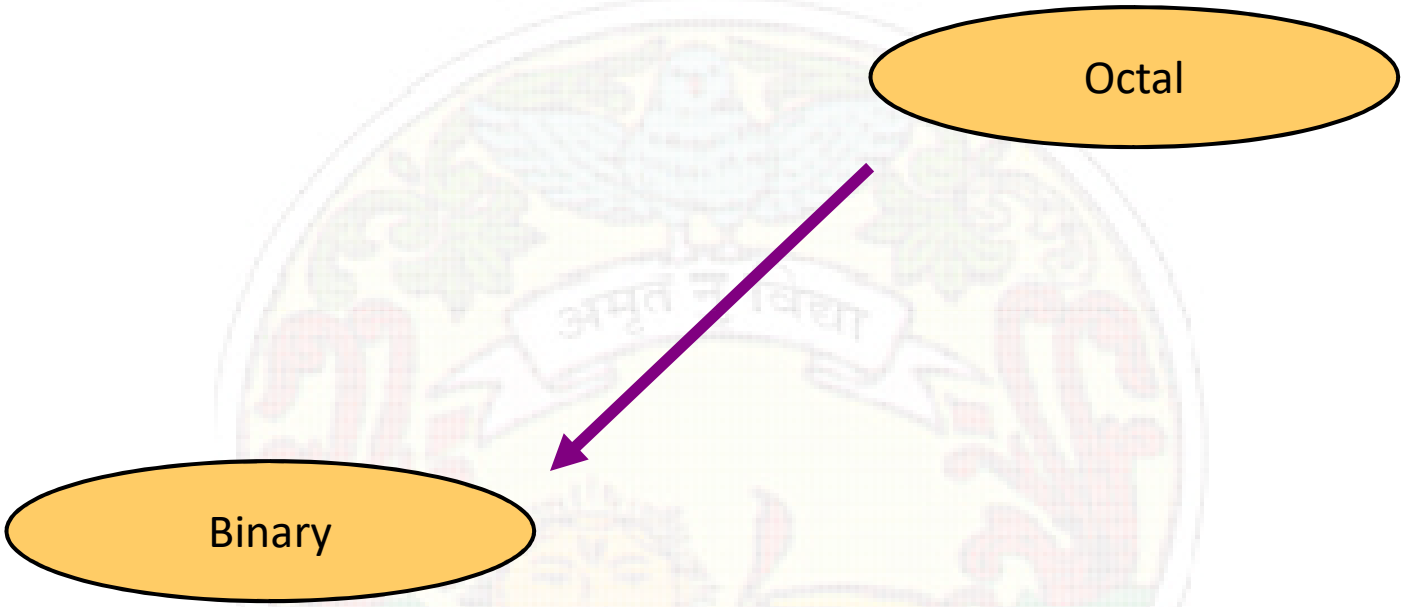
Example

$125_{10} = ?_2$

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

$125_{10} = 1111101_2$

Octal to Binary

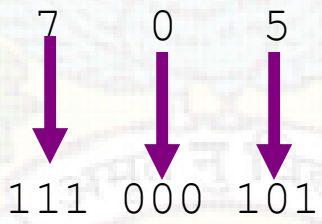


Octal to Binary

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

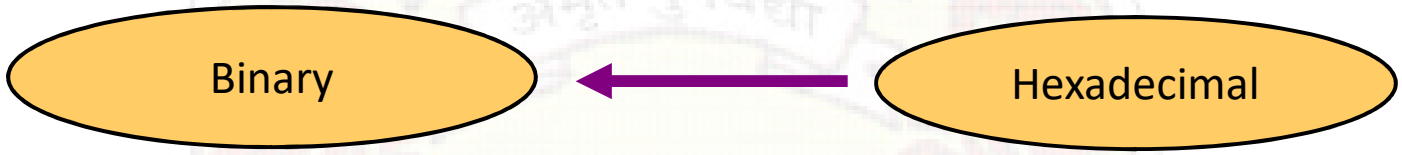
Example

$705_8 = ?_2$



$705_8 = 111000101_2$

Hexadecimal to Binary

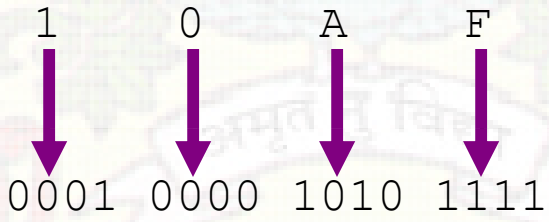


Hexadecimal to Binary

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

Example

$10AF_{16} = ?_2$



$10AF_{16} = 0001000010101111_2$

Decimal to Octal

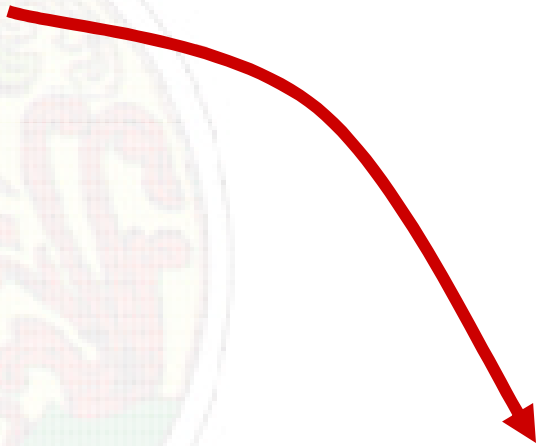
Decimal to Octal

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

Example

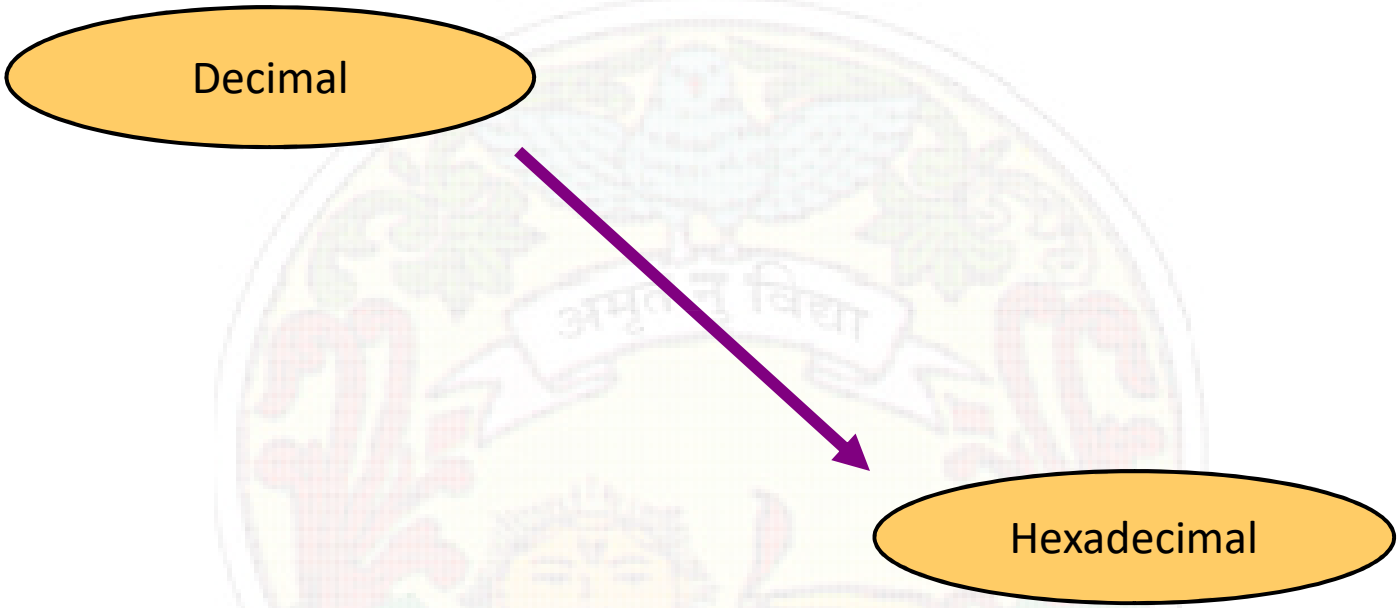
$1234_{10} = ?_8$

8	1234	
	154	2
8		
	19	2
8		
	2	3
8		
	0	2



$1234_{10} = 2322_8$

Decimal to Hexadecimal



Decimal to Hexadecimal

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

Example

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

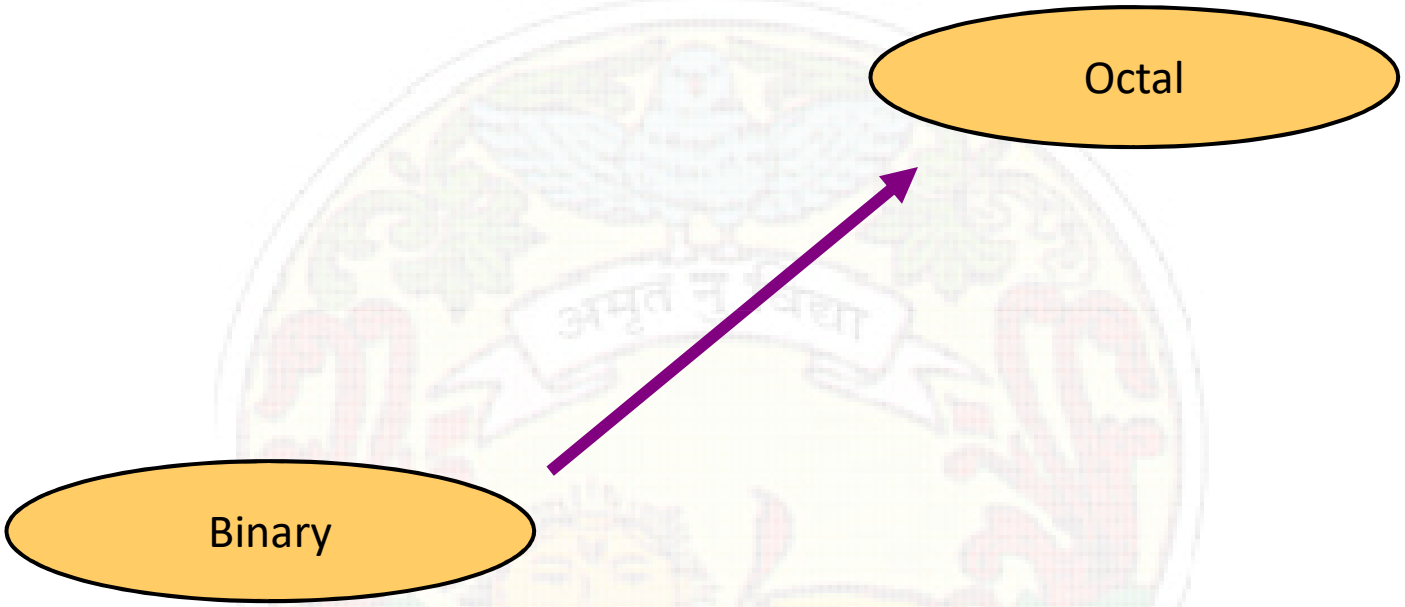
16 | 1234
// 2

16 | 4 13 = D

16 | 0 4

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

Binary to Octal

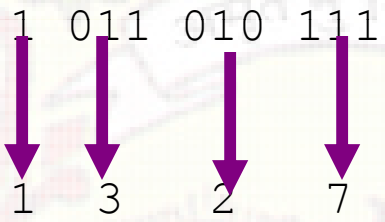


Binary to Octal

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

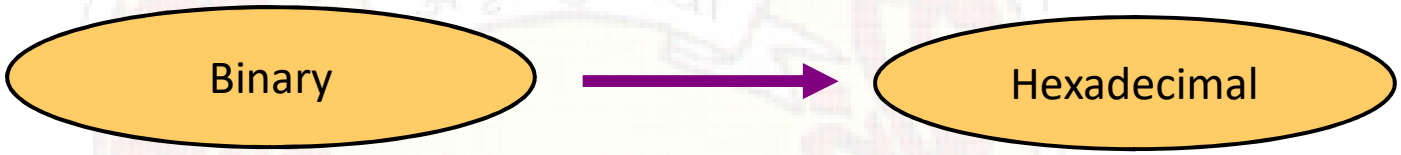
Example

$1011010111_2 = ?_8$



$1011010111_2 = 1327_8$

Binary to Hexadecimal

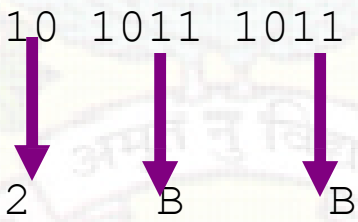


Binary to Hexadecimal

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

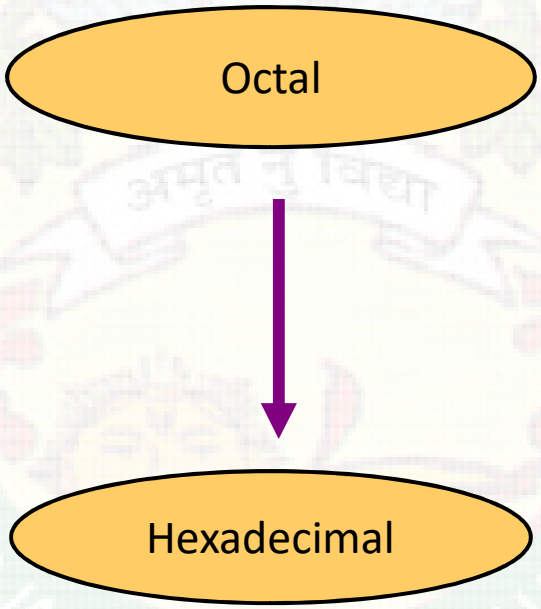
Example

$1010111011_2 = ?_{16}$



$1010111011_2 = 2BB_{16}$

Octal to Hexadecimal

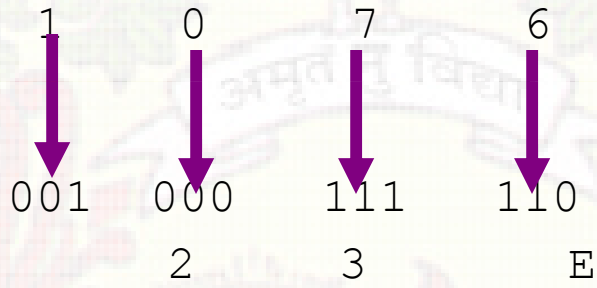


Octal to Hexadecimal

- Technique
  - Use binary as an intermediary

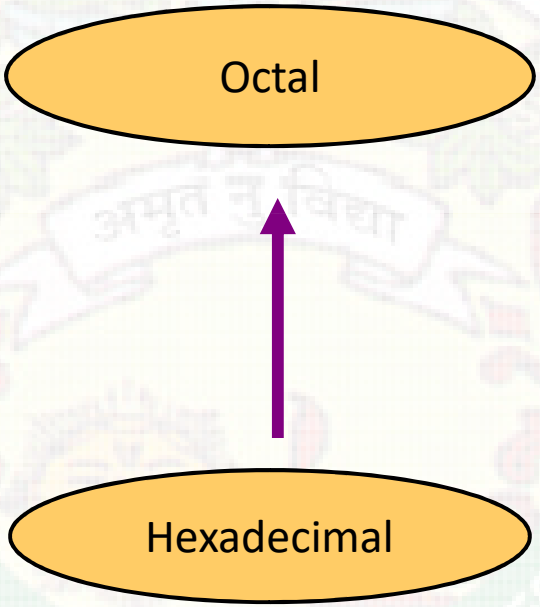
Example

$1076_8 = ?_{16}$



$1076_8 = 23E_{16}$

Hexadecimal to Octal

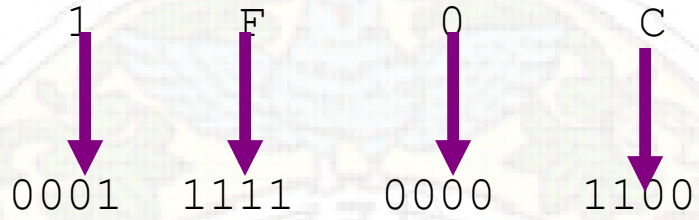


Hexadecimal to Octal

- Technique
  - Use binary as an intermediary

Example

$1F0C_{16} = ?_8$



$1F0C_{16} = 17414_8$

Thank you