

# Number Systems

# Common Number Systems

System	Base	Symbols	Used by humans?	Used in computers?
Decimal	10	0, 1, ... 9	Yes	No
Binary	2	0, 1	No	Yes
Octal	8	0, 1, ... 7	No	No
Hexa-decimal	16	0, 1, ... 9, A, B, ... F	No	No

# 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

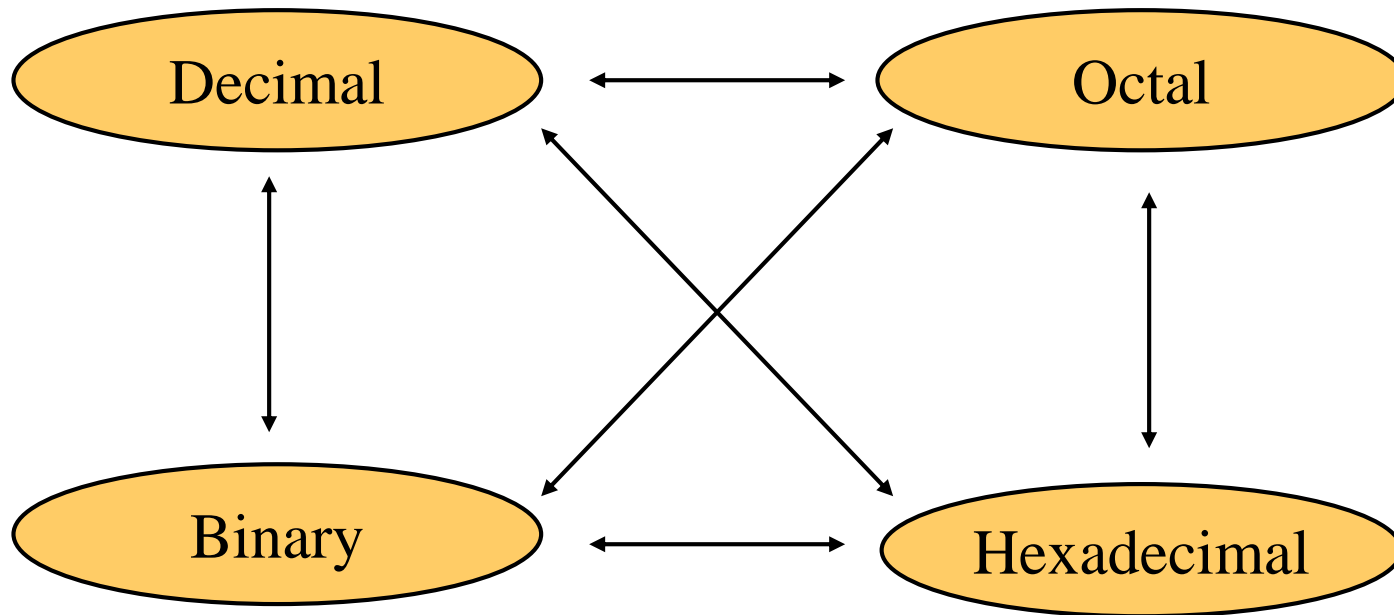
# Quantities/Counting (3 of 3)

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.

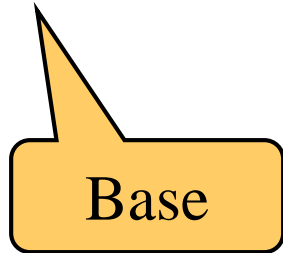
# Conversion Among Bases

- The possibilities:

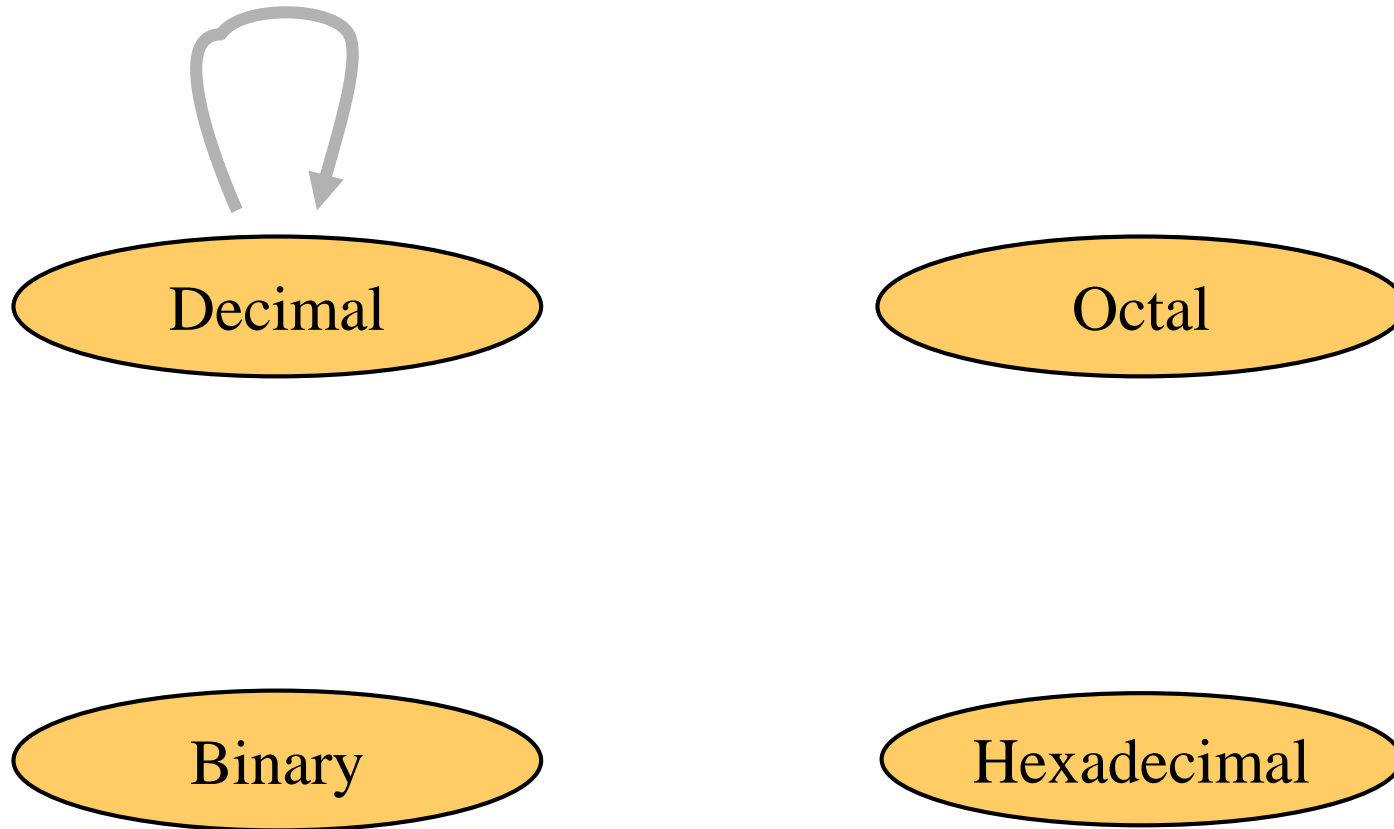


# Quick Example

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



# Decimal to Decimal (just for fun)



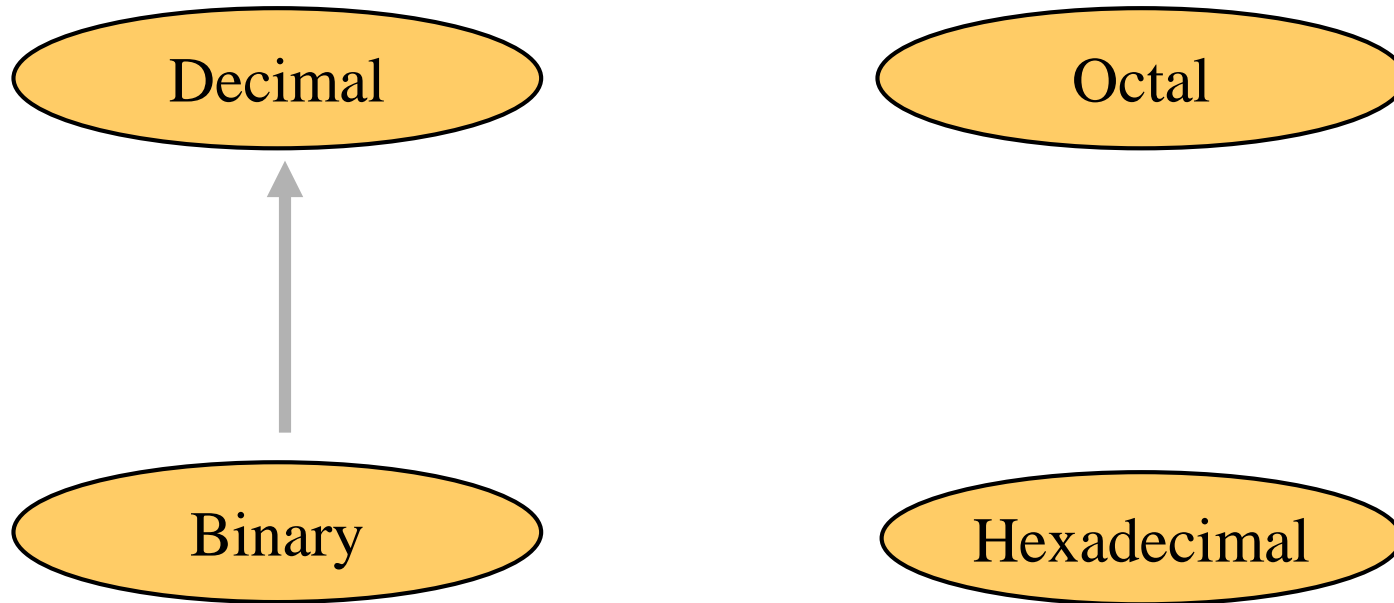
Next slide...

$$125_{10} \Rightarrow \begin{array}{r} 5 \times 10^0 = 5 \\ 2 \times 10^1 = 20 \\ 1 \times 10^2 = 100 \\ \hline 125 \end{array}$$

Weight

Base

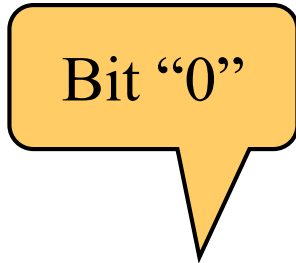
# Binary to Decimal



# 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

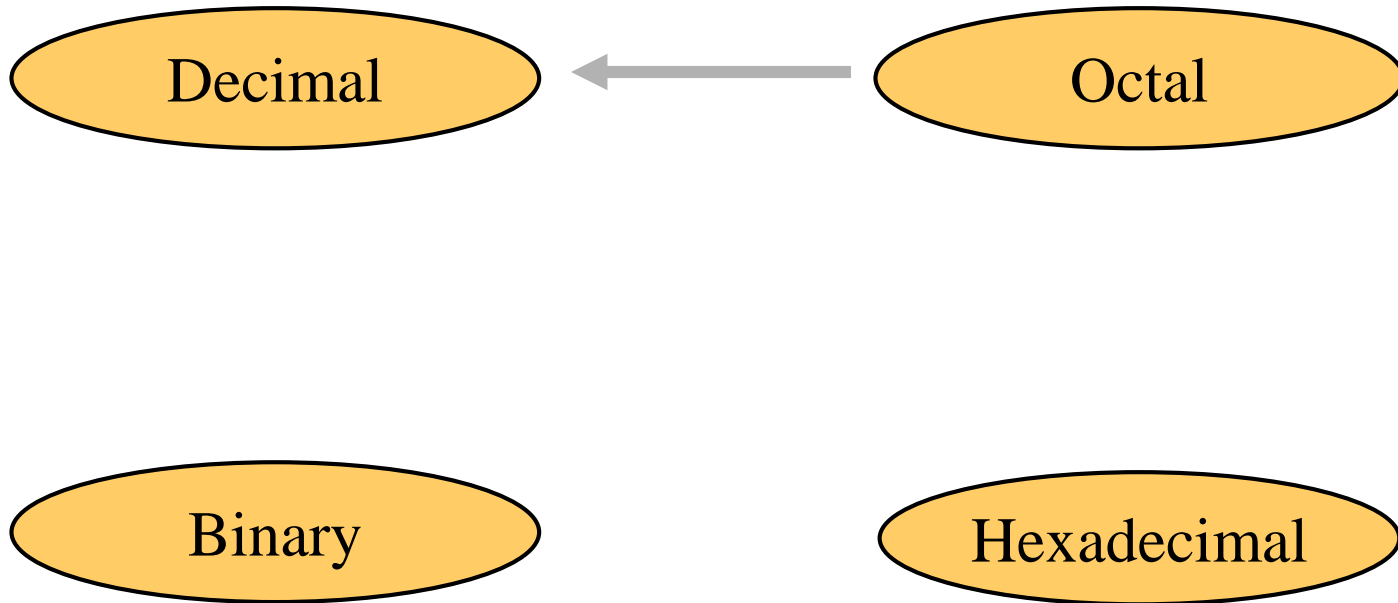
# Example



$101011_2 \Rightarrow$

1	x	$2^0$	=	1
1	x	$2^1$	=	2
0	x	$2^2$	=	0
1	x	$2^3$	=	8
0	x	$2^4$	=	0
1	x	$2^5$	=	32
				<hr/>
				$43_{10}$

# Octal to Decimal



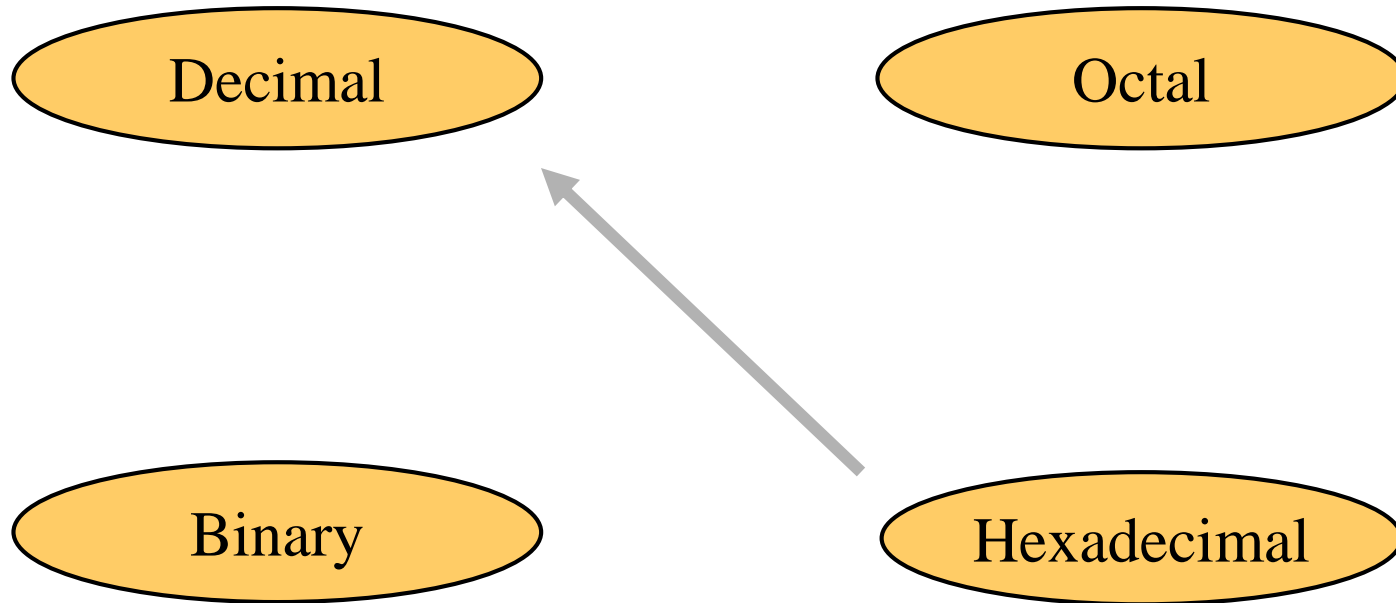
# Octal to Decimal

- 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

$$\begin{array}{r} 724_8 \Rightarrow \\ 4 \times 8^0 = 4 \\ 2 \times 8^1 = 16 \\ 7 \times 8^2 = 448 \\ \hline 468_{10} \end{array}$$

# 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

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

# Decimal to Binary

Decimal

Octal



Binary

Hexadecimal

# 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.

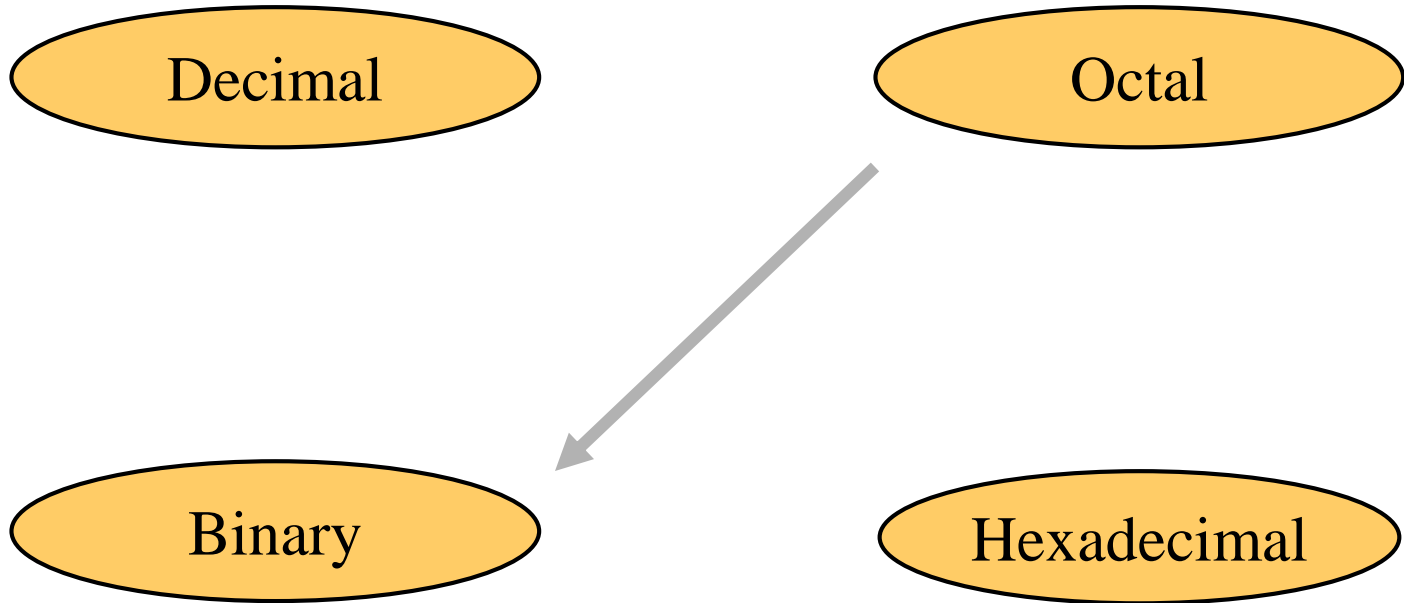
# Example

$$125_{10} = ?_2$$

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

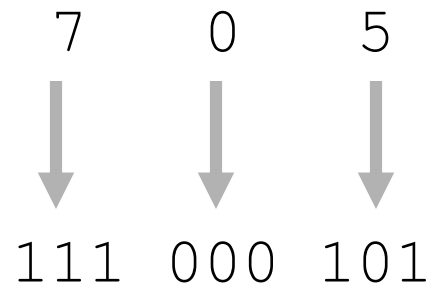


# Octal to Binary

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

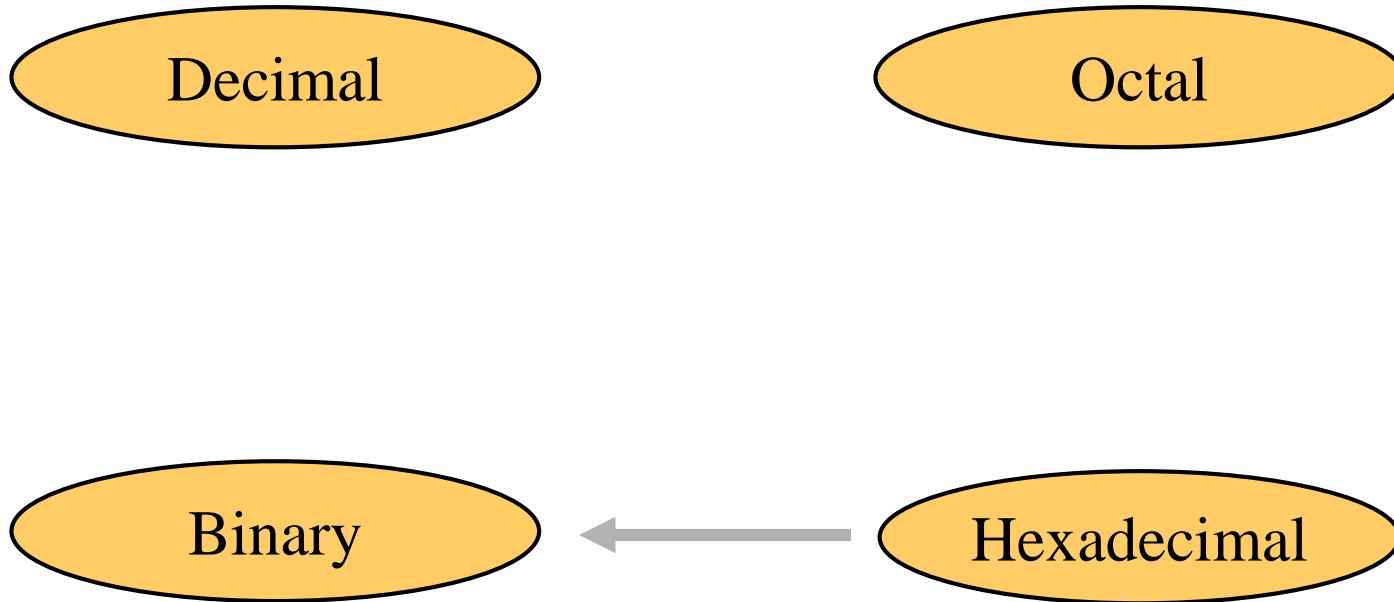
# Example

$$705_8 = ?_2$$



$$705_8 = 111000101_2$$

# Hexadecimal to Binary

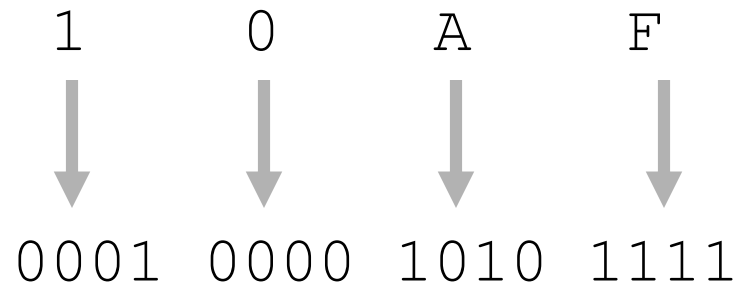


# Hexadecimal to Binary

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

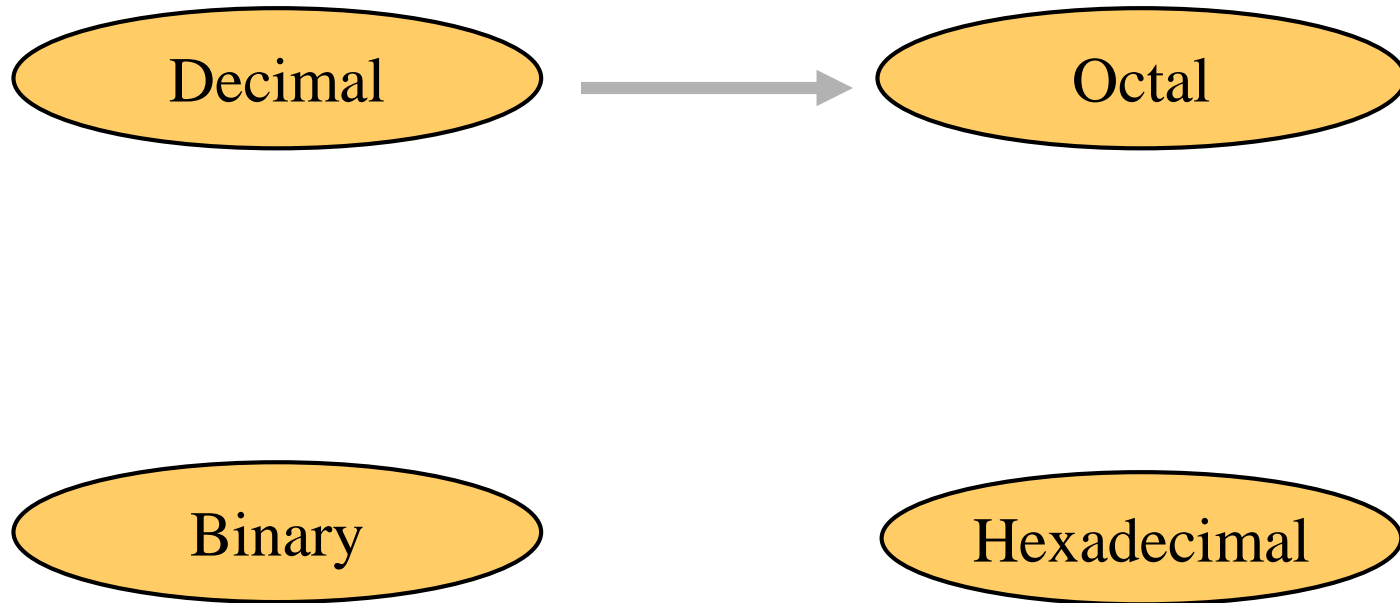
# Example

$$10AF_{16} = ?_2$$



$$10AF_{16} = 0001000010101111_2$$

# Decimal to Octal



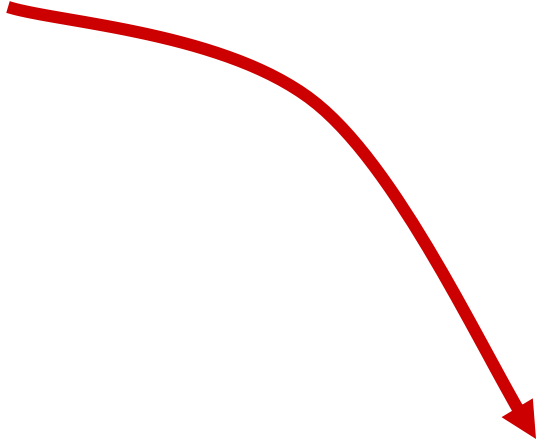
# Decimal to Octal

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

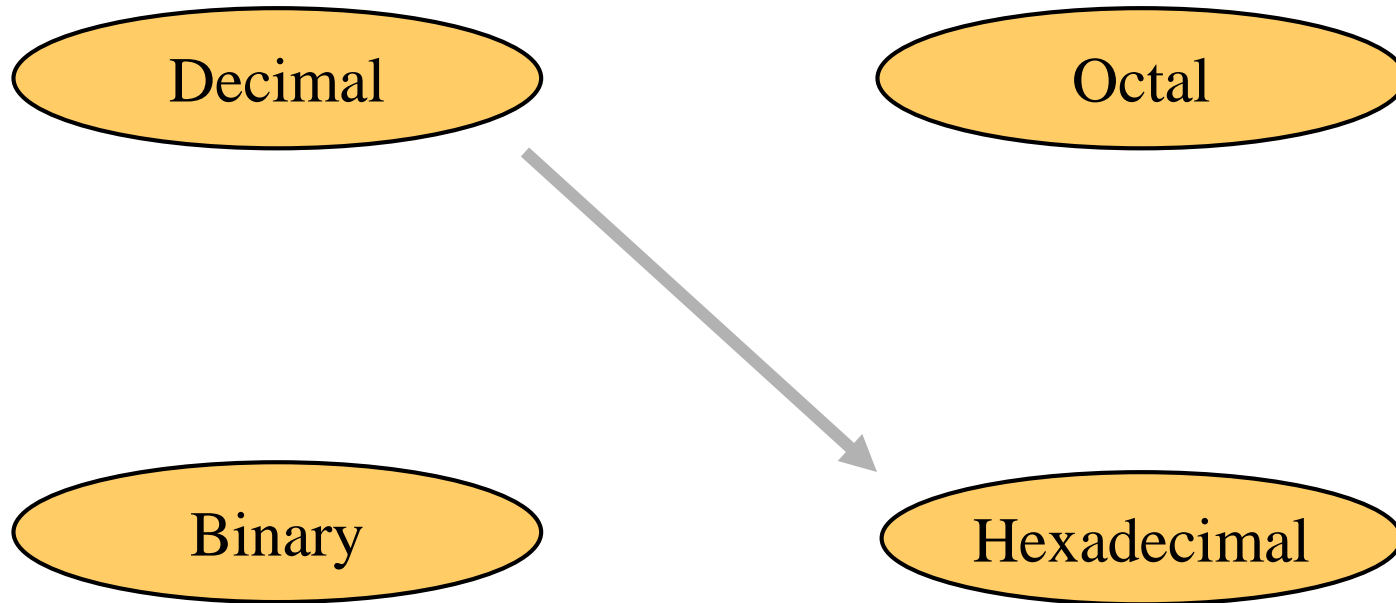
# Example

$$1234_{10} = ?_8$$

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


$$1234_{10} = 2322_8$$

# Decimal to Hexadecimal



# Decimal to Hexadecimal

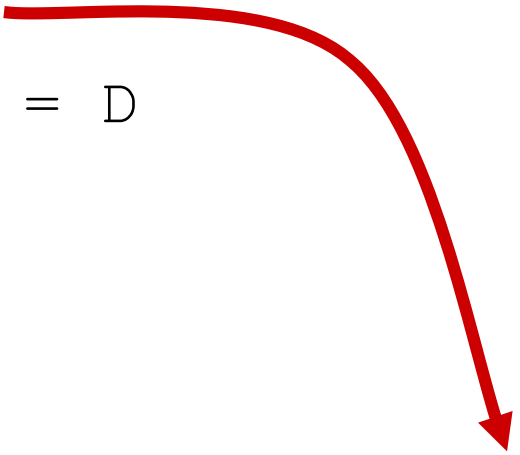
- Technique
  - Divide by 16
  - Keep track of the remainder

# Example

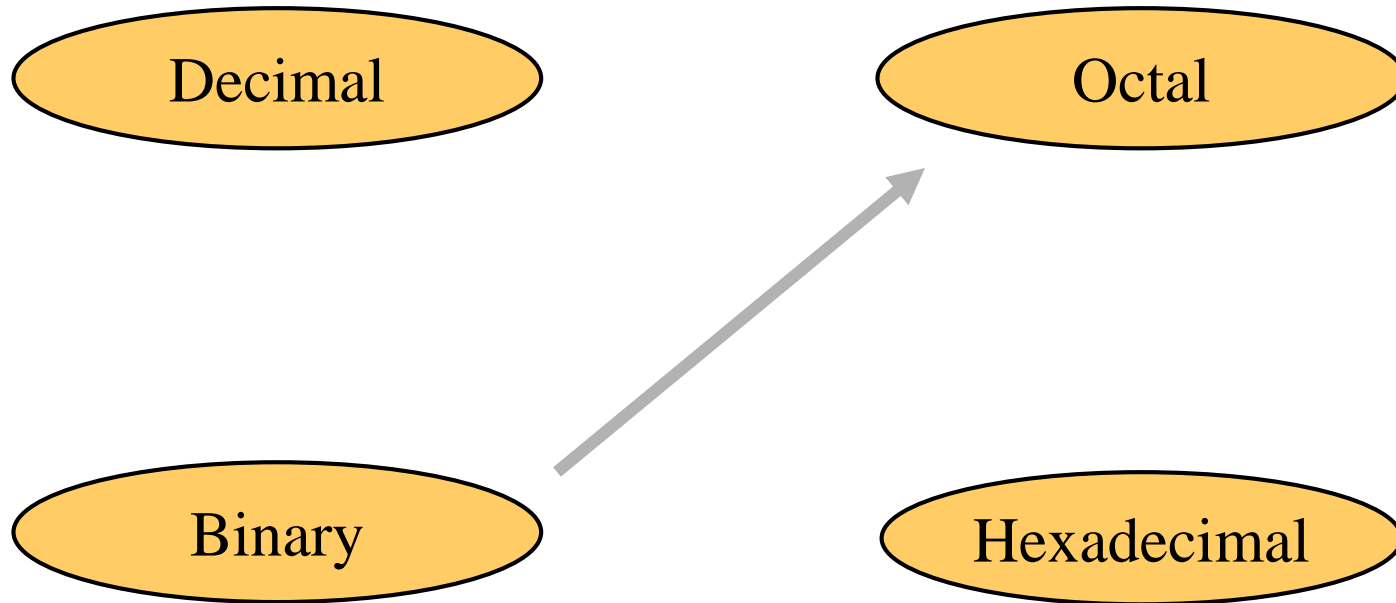
$$1234_{10} = ?_{16}$$

$$\begin{array}{r|l} 16 & 1234 \\ \hline 16 & \phantom{1}77 \\ \hline 16 & \phantom{1}\phantom{7}4 \\ \hline & 0 \end{array}$$

$$\begin{array}{l} 2 \\ 13 = D \\ 4 \end{array}$$


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

# Binary to Octal

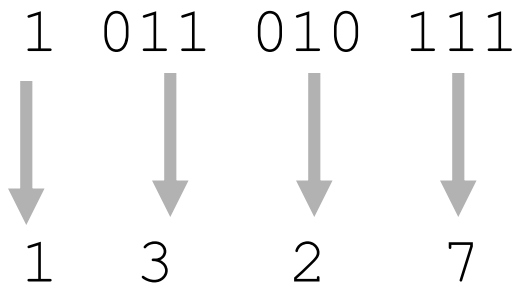


# Binary to Octal

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

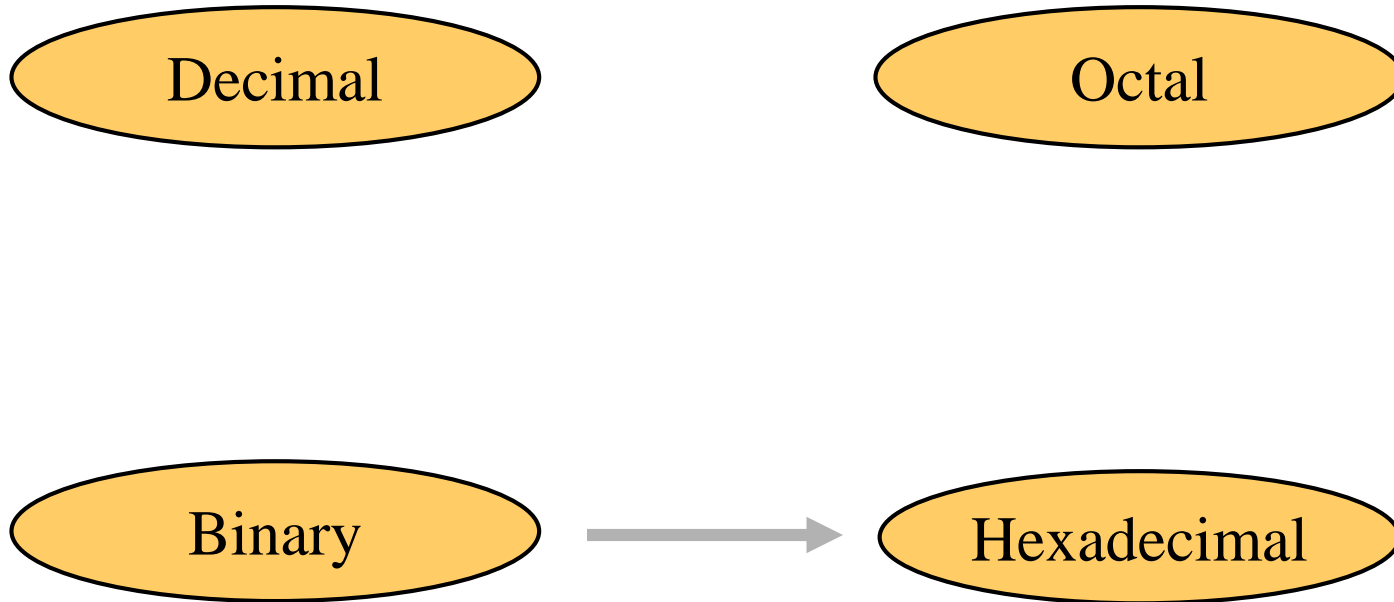
# Example

$$1011010111_2 = ?_8$$



$$1011010111_2 = 1327_8$$

# Binary to Hexadecimal

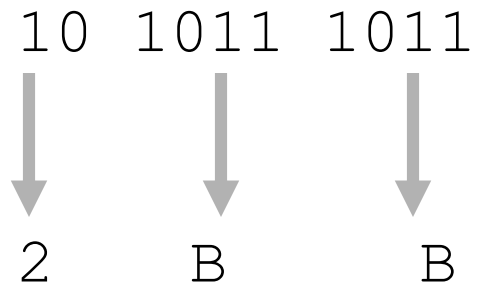


# Binary to Hexadecimal

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

# Example

$$1010111011_2 = ?_{16}$$



$$1010111011_2 = 2BB_{16}$$

# Octal to Hexadecimal

Decimal

Octal

Binary

Hexadecimal

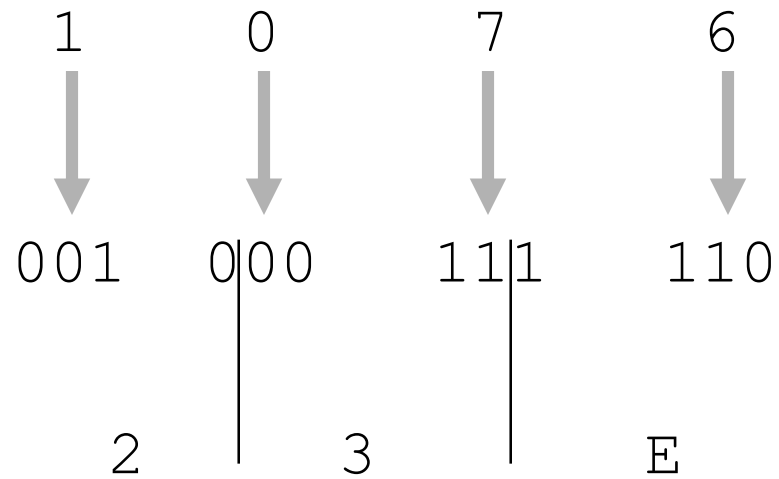


# Octal to Hexadecimal

- Technique
  - Use binary as an intermediary

# Example

$$1076_8 = ?_{16}$$



$$1076_8 = 23E_{16}$$

# Hexadecimal to Octal

Decimal

Octal

Binary

Hexadecimal

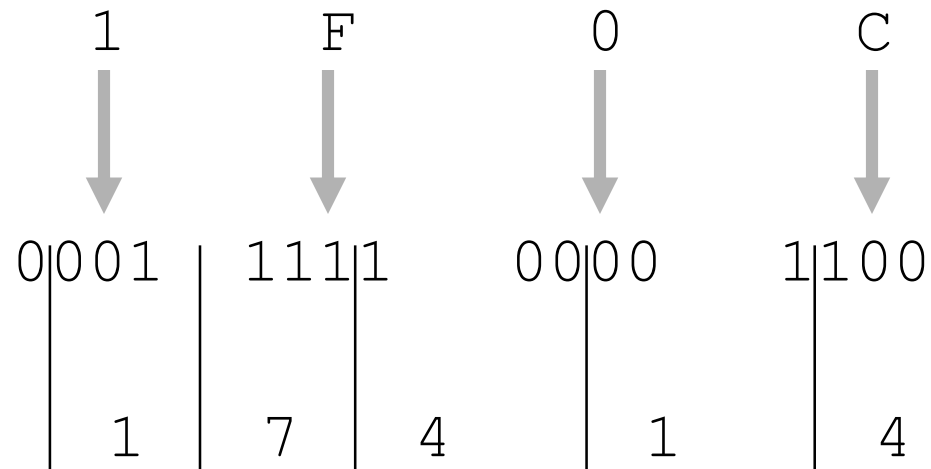


# Hexadecimal to Octal

- Technique
  - Use binary as an intermediary

# Example

$$1F0C_{16} = ?_8$$



$$1F0C_{16} = 17030_8$$

# Exercise – Convert ...

Decimal	Binary	Octal	Hexa- decimal
33			
	1110101		
		703	
			1AF

Don't use a calculator!

Skip answer

Answer

# Exercise – Convert ...

Answer

Decimal	Binary	Octal	Hexa- decimal
33	100001	41	21
117	1110101	165	75
451	111000011	703	1C3
431	110101111	657	1AF



# Common Powers (1 of 2)

- Base 10

Power	Preface	Symbol	Value
$10^{-12}$	pico	p	.000000000001
$10^{-9}$	nano	n	.000000001
$10^{-6}$	micro	$\mu$	.000001
$10^{-3}$	milli	m	.001
$10^3$	kilo	k	1000
$10^6$	mega	M	1000000
$10^9$	giga	G	1000000000
$10^{12}$	tera	T	1000000000000

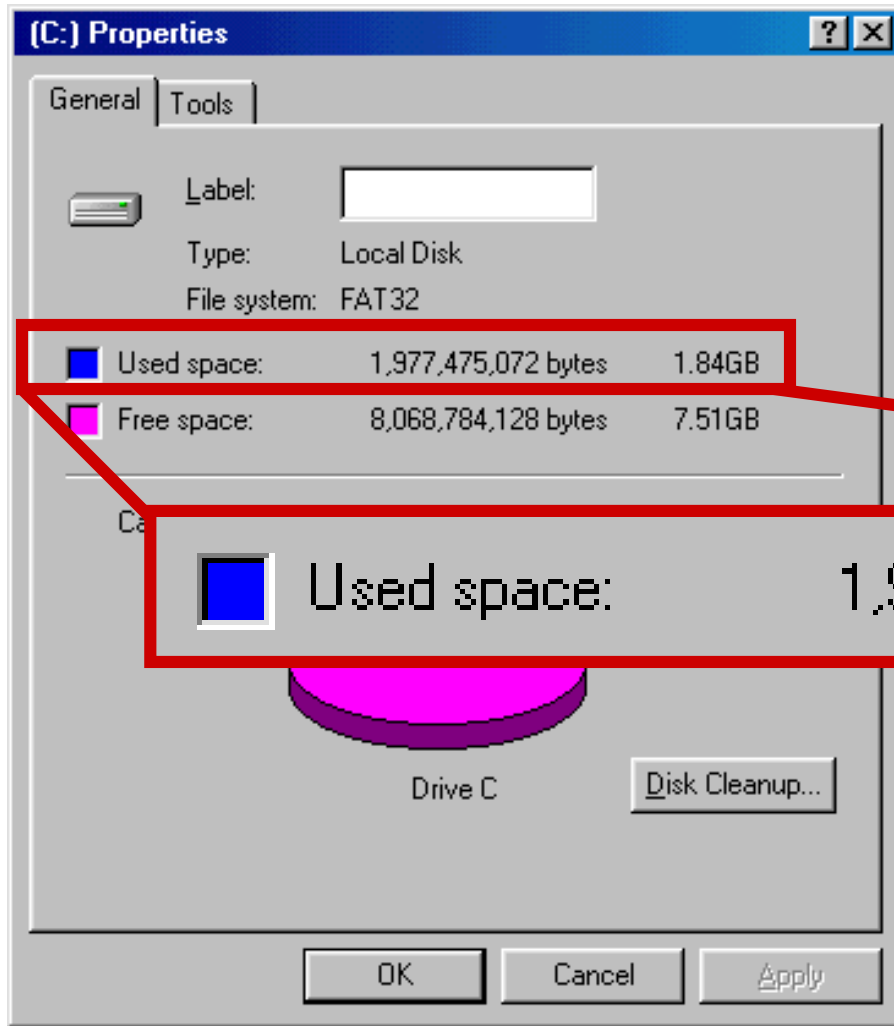
# Common Powers (2 of 2)

- Base 2

Power	Preface	Symbol	Value
$2^{10}$	kilo	k	1024
$2^{20}$	mega	M	1048576
$2^{30}$	Giga	G	1073741824

- What is the value of “k”, “M”, and “G”?
- In computing, particularly w.r.t. memory, the base-2 interpretation generally applies

# Example



In the lab...

1. Double click on My Computer
2. Right click on C:
3. Click on Properties

Used space: 1,977,475,072 bytes 1.84GB

$$/ 2^{30} =$$

# Exercise – Free Space

- Determine the “free space” on all drives on a machine in the lab

Drive	Free space	
	Bytes	GB
A:		
C:		
D:		
E:		
etc.		

# Review – multiplying powers

- For common bases, add powers

$$a^b \times a^c = a^{b+c}$$

$$2^6 \times 2^{10} = 2^{16} = 65,536$$

or...

$$2^6 \times 2^{10} = 64 \times 2^{10} = 64\text{k}$$

Thank you