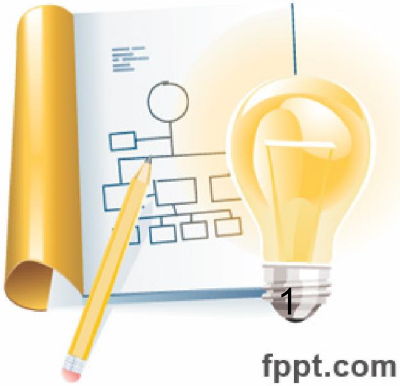
2. Number Systems



Number Systems

- A number system defines a set of values used to represent quantity.
- It can be categorized in two broad categories:

1. **Non – Positional Number Systems**

➢ Stones or sticks were used to indicate values

➢ Difficult to perform arithmetic because it has no symbol for zero.

Example: Roman number systems – Few characters are used such as I, V, X, L (fifty), C (hundred)

Number System

2. Positional Number Systems

- The value of each digit in number is defined not only by the symbol but also by symbol's position.
- Positional number systems have base or radix.

Common Number Systems

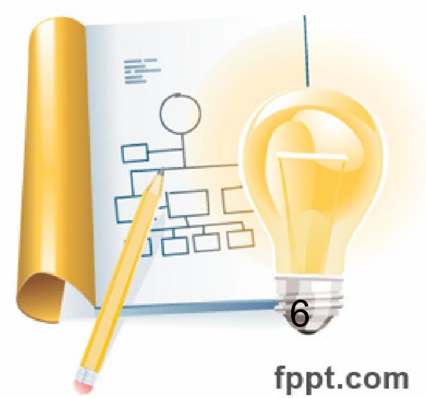
System	Base	Symbols	Used by humans?	Used in computers?
Decimal	10	0, 1, … 9	Yes	No
Binary	2	0, 1	No	Yes
Octal	8	0, 1, … 7	No	No
Hexa-decimal	16	0, 1, … 9, A, B, … F	No	No

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)

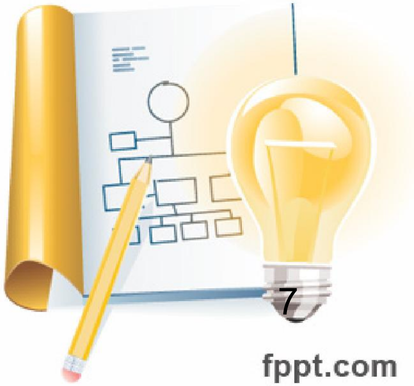
Decimal	Binary	Octal	Hexa-decimal
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F



Quantities/Counting (3 of 3)

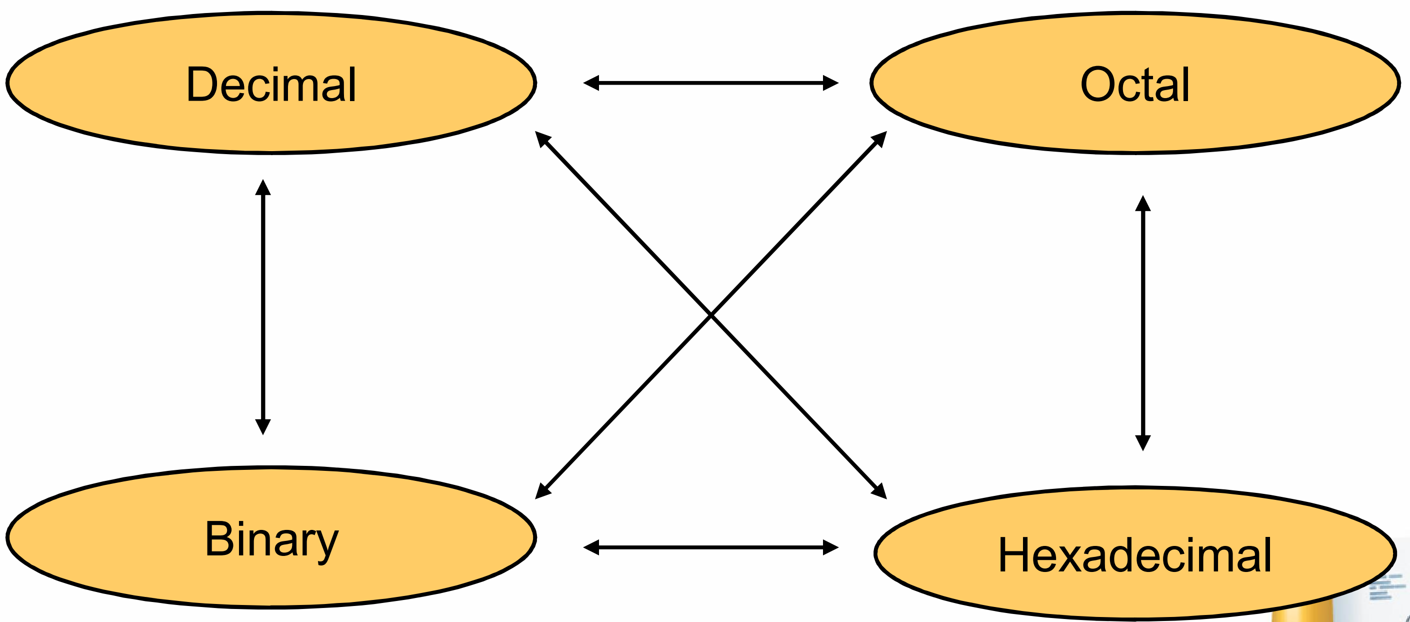
Decimal	Binary	Octal	Hexa-decimal
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
22	10110	26	16
23	10111	27	17

Etc.



Conversion Among Bases

- The possibilities:

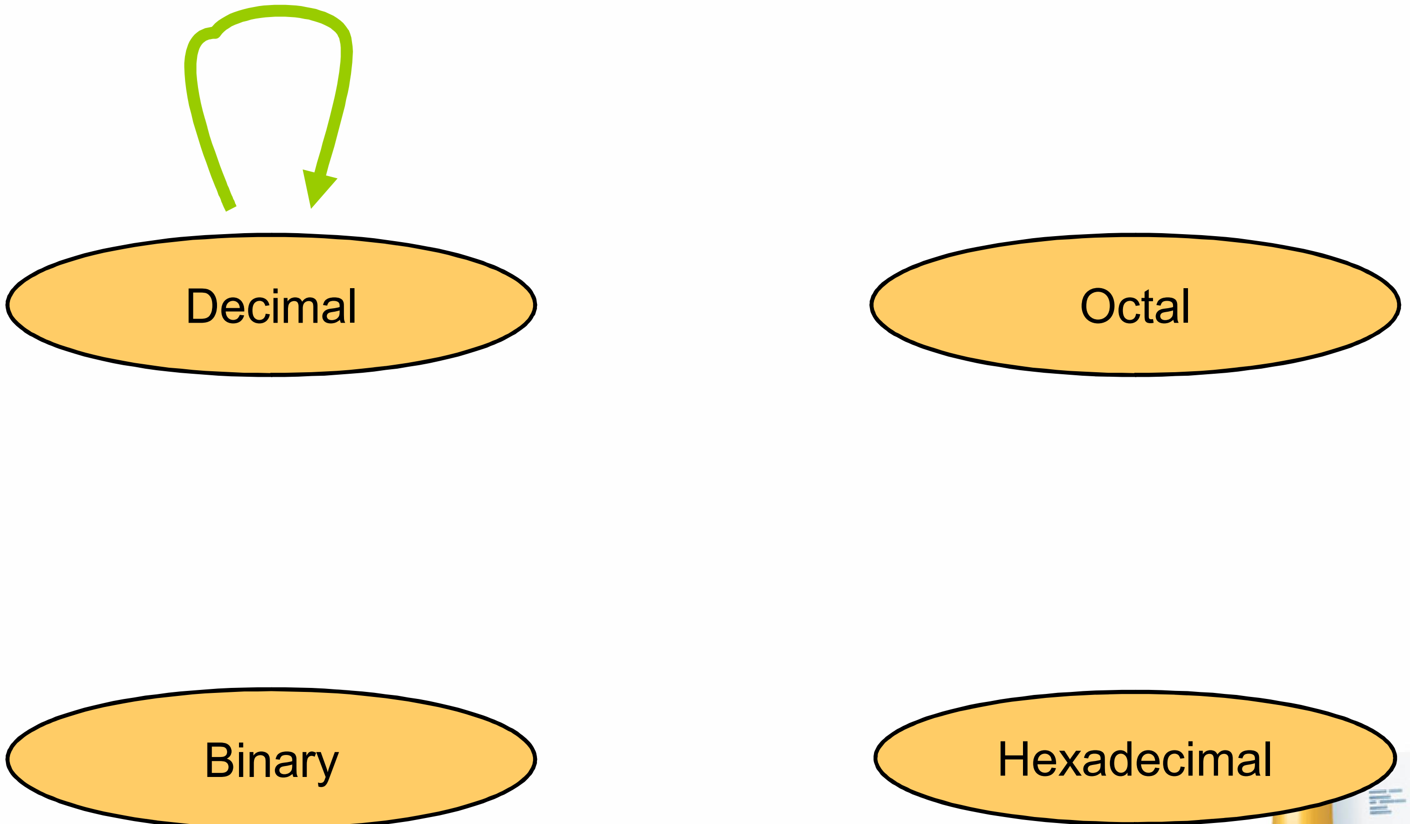


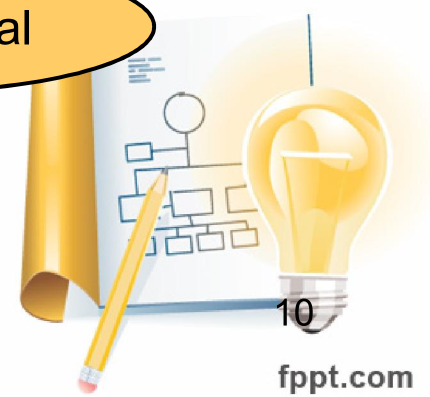
Quick Example

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

Base

Decimal to Decimal (just for fun)





Weight

125_{10} =>

$5 \times 10^0 = 5$

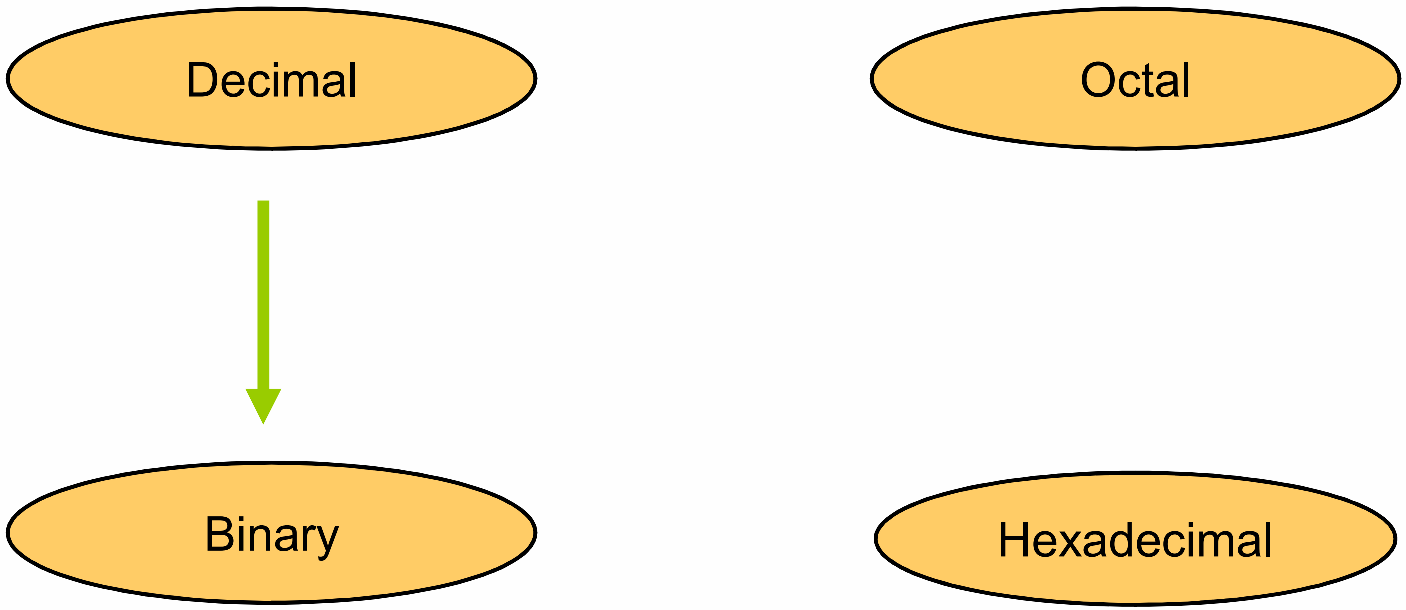
$2 \times 10^1 = 20$

$1 \times 10^2 = 100$

125

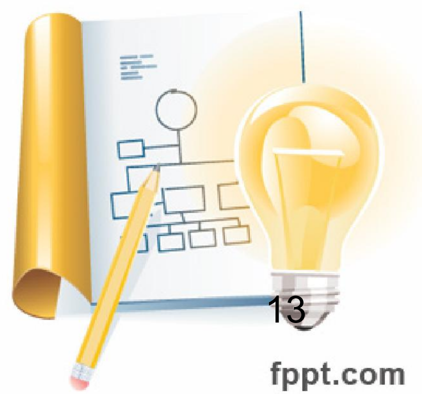
Base

Decimal to Binary



Decimal to Binary

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



Conversion of Decimal to Binary

1. Determine the Binary equivalent of (125)10

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

Least Significant Bit (LSB)

Most Significant Bit (MSB)

$$125_{10} = 1111101_2$$

Conversion of Decimal to Binary

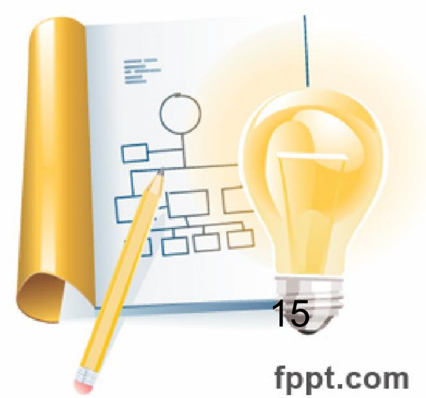
2. Determine the Binary equivalent of (36)10

2	36	Remainder
2	18	0
2	9	0
2	4	1
2	2	0
2	1	0
	0	1

Least Significant Bit (LSB)

Most Significant Bit (MSB)

$$36_{10} = 100100_2$$



Conversion of Decimal to Binary

3. Determine the Binary equivalent of (671)10

2	671	Remainder
2	335	1
2	167	1
2	83	1
2	41	1
2	20	1
2	10	0
2	5	0
2	2	1
2	1	0
	0	1

Least Significant Bit (LSB)

Most Significant Bit (MSB)

$$671_{10} = 1010011111_{2}$$

Conversion of Decimal Fraction to Binary Fraction

1. Determine the Binary equivalent of (0.375)10

0.375	**X**	**2**	**= 0.750**	**0**
0.750	X	2	= 1.500	1
0.500	X	2	= 1.000	1

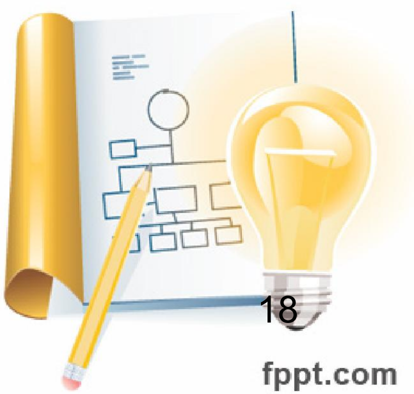
$$0.375_{10} = 011_2$$

Conversion of Decimal Fraction to Binary Fraction

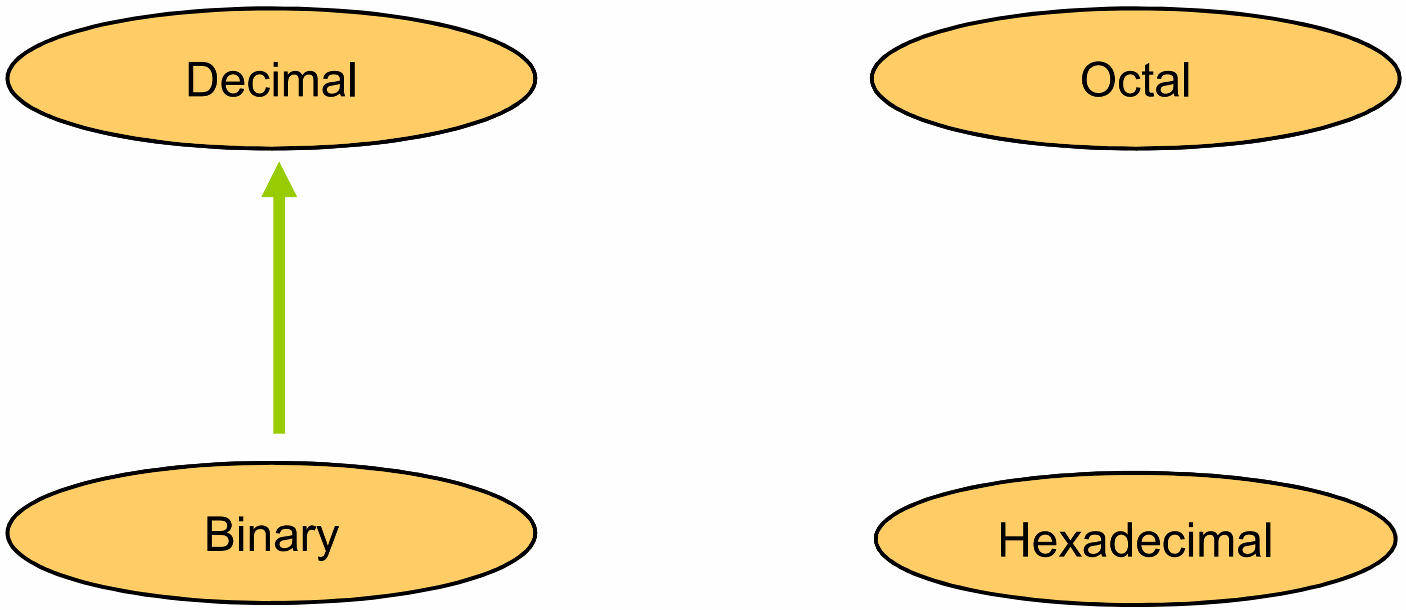
2. Determine the Binary equivalent of (0.29)10

0.29	**X**	**2**	**=**	**0.58**	**0**
0.58	X	2	=	1.16	1
0.16	X	2	=	0.32	0
0.32	X	2	=	0.64	0
0.64	X	2	=	1.28	1
0.28	X	2	=	0.56	0
	.				
	.				
	.				
	∞				

$$0.29_{10} = (0.010010)_2$$



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

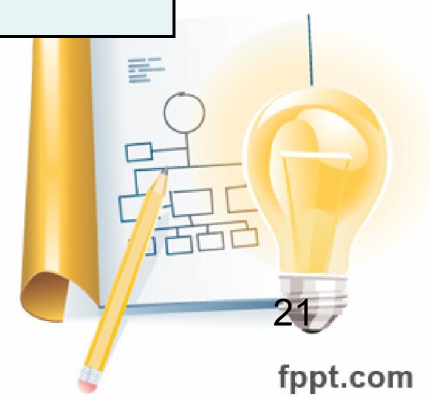
Binary to Decimal

1. Determine the Decimal equivalent of (101011)2

Binary Number	**1**	**0**	**1**	**0**	**1**	**1**
Weight of Each Bit	2^5	2^4	2^3	2^2	2^1	2^0
Weighted Value	1 X 2^5	0 X 2^4	1 X 2^3	0 X 2^2	1 X 2^1	1 X 2^0
Solved Multiplication	32	0	8	0	2	1

Sum of weight of all bits = 32 + 0 + 8 + 2 + 1

= 43



Binary to Decimal

2. Determine the Decimal equivalent of (11010)2

Binary Number	**1**	**1**	**0**	**1**	**0**
Weight of Each Bit	2^4	2^3	2^2	2^1	2^0
Weighted Value	1 X 2^4	1 X 2^3	0 X 2^2	1 X 2^1	0 X 2^0
Solved Multiplication	16	8	0	2	0

Sum of weight of all bits = 16 + 8 + 0 + 2 + 0
= 26

Binary to Decimal

3. Determine the Decimal equivalent of (10110011)2

Binary Number	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**1**
Weight of Each Bit	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Weighted Value	1 X 2^7	0 X 2^6	1 X 2^5	1 X 2^4	0 X 2^3	0 X 2^2	1 X 2^1	1 X 2^0
Solved Multiplication	128	0	32	16	0	0	2	1

Sum of weight of all bits = 128 + 0 + 32 + 16 + 0 + 0 + 2 +1

= 179

Conversion of Binary Fraction to Decimal Fractions

1. Determine the Decimal equivalent of (0.01101)2

Binary Number	0	1	1	0	1
Weight of Each Bit	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
Weighted Value	0 X 2^{-1}	1 X 2^{-2}	1 X 2^{-3}	0 X 2^{-4}	1 X 2^{-5}
Solved Multiplication	0	1/4	1/8	0	1/32

Sum of weight of all bits = 0 + 1/4 + 1/8 + 0 + 1/32
= 0 + 0.25 + 0.125 + 0 + 0.03125
= 0.40625

Conversion of Binary Fraction to Decimal Fractions

2. Determine the Decimal equivalent of (11101.10111)2

Binary Number	**1**	**1**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**1**
Weight of Each Bit	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
Weighted Value	1 X 2^4	1 X 2^3	1 X 2^2	0 X 2^1	1 X 1	1 X 2^{-1}	0 X 2^{-2}	1 X 2^{-3}	1 X 2^{-4}	1 X 2^{-5}
Solved Multiplication	16	8	4	0	1	1/2	0	1/8	1/16	1/32

Sum of weight of all bits = 16+8+4+0+1+(1/2)+0+(1/8)+(1/16)+(1/32)
= 16+8+4+0+1+0.5+0+ 0.125+0.0625+0.03125
= 29.71875

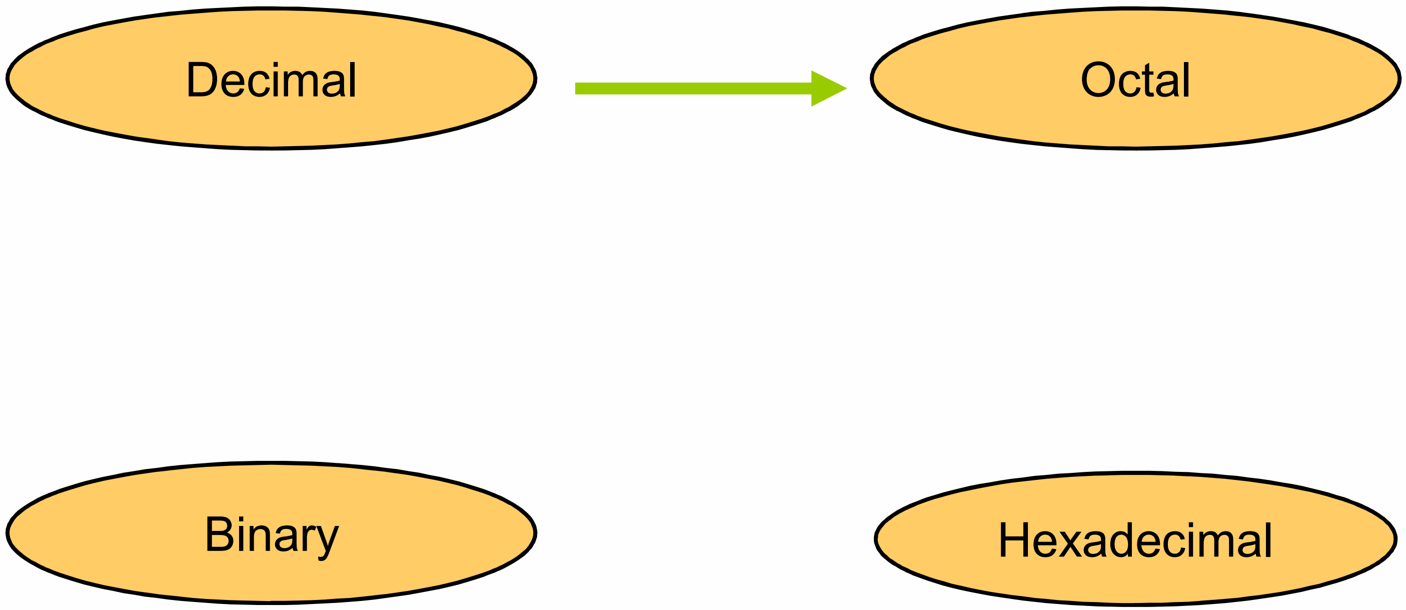
Conversion of Binary Fraction to Decimal Fractions

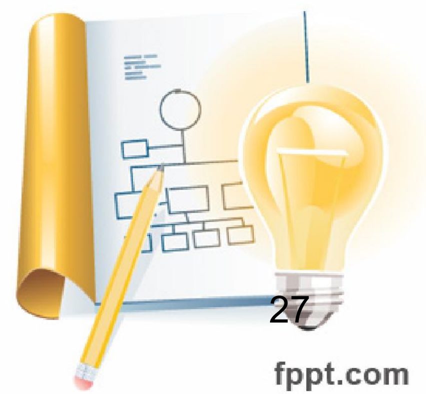
3. Determine the Decimal equivalent of (10.1011)2

Binary Number	**1**	**0**	**1**	**0**	**1**	**1**
Weight of Each Bit	2^{1}	2^{0}	2^{-1}	2^{-2}	2^{-3}	2^{-4}
Weighted Value	1 X 2	1 X 2^{0}	1 X 2^{-1}	0 X 2^{-2}	1 X 2^{-3}	1 X 2^{-4}
Solved Multiplication	2	1	1/2	0	1/8	1/16

Sum of weight of all bits = 2 + 1 + (1/2) + 0+ (1/8) + (1/16)
=2 + 1 + 0.5 + 0 + 0.125 + 0.0625
= 2.6875

Decimal to Octal





Decimal to Octal

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

Conversion of Decimal to Octal

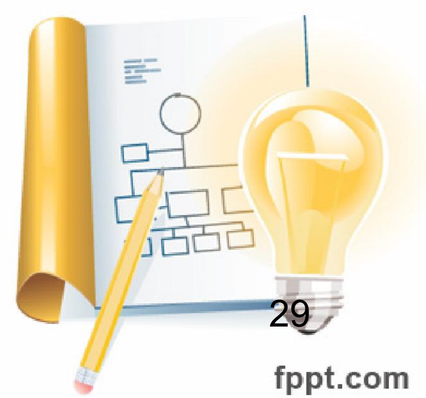
1. Determine the Octal equivalent of (1234)10

8	**1234**	**Remainder**
8	154	2
8	19	2
8	2	3
	0	2

Least Significant Bit (LSB)

Most Significant Bit (MSB)

$$1234_{10} = 2322_{8}$$



Conversion of Decimal to Octal

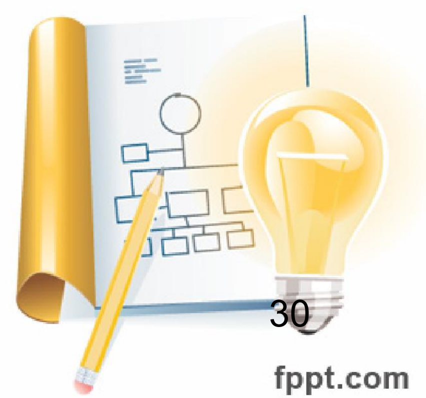
2. Determine the Octal equivalent of (359)10

8	359	Remainder
8	44	7
8	5	4
	0	5

Least Significant Bit (LSB)

Most Significant Bit (MSB)

$$359_{10} = 547_{8}$$



Conversion of Decimal to Octal

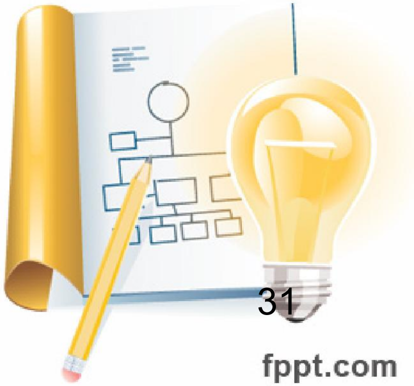
3. Determine the Octal equivalent of (432267)10

8	432267	Remainder
8	54033	3
8	6754	1
8	844	2
8	105	4
8	13	1
8	1	5
	0	1

Least Significant Bit (LSB)

Most Significant Bit (MSB)

$$432267_{10} = 1514213_8$$



Conversion of Decimal Fraction to Octal Fraction

1. Determine the Octal equivalent of (0.3125)10

0.3125	**X**	**8**	**= 2.5**	**2**
0.50	X	8	= 4.0	4

$$0.375_{10} = 0.24_8$$

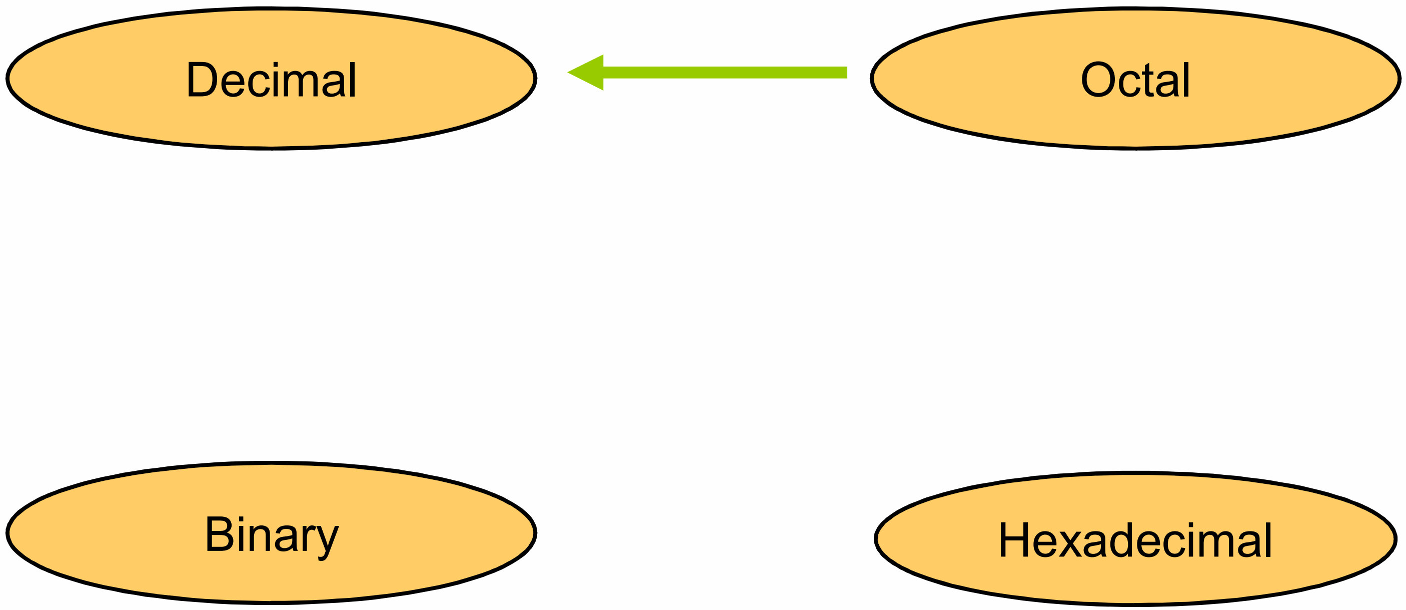
Conversion of Decimal Fraction to Octal Fraction

2. Determine the Octal equivalent of (0.1325)10

0.1325	**X**	**8**	**=**	**1.0600**	**1**
0.0600	X	8	=	0.4800	0
0.4800	X	8	=	3.8400	3
0.8400	X	8	=	6.7200	6
6.7200	X	8	=	5.7600	5
	.				
	.				
	.				
	∞				

$$0.1325_{10} = 0.10365_{8}$$

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

Conversion of Octal to Decimal

1. Determine the Decimal equivalent of (724)8

Octal Number	7	2	4
Weight of Each Bit	8^2	8^1	8^0
Weighted Value	7 X 8^2	2 X 8^1	4 X 8^0
Solved Multiplication	448	16	4

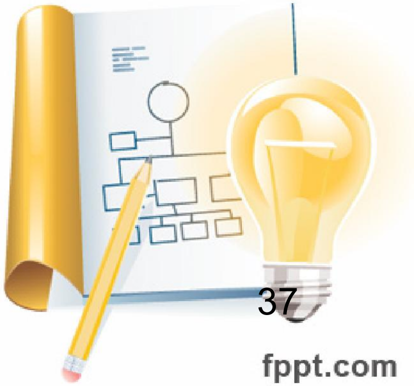
Sum of weight of all bits = 4 + 16 + 448
= 468

Conversion of Octal to Decimal

2. Determine the Decimal equivalent of (456)8

Octal Number	**4**	**5**	**6**
Weight of Each Bit	8^2	8^1	8^0
Weighted Value	4 X 8^2	5 X 8^1	6 X 8^0
Solved Multiplication	256	40	6

Sum of weight of all bits = 256 + 40 + 6
= 302

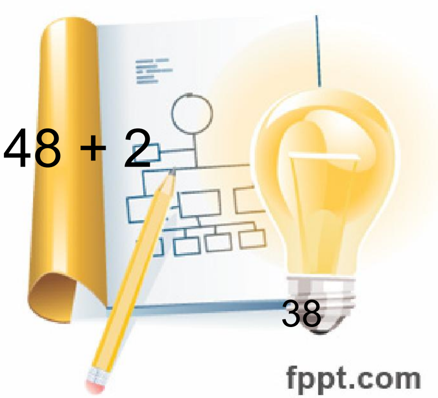


Conversion of Octal to Decimal

3. Determine the Decimal equivalent of (127662)8

Octal Number	**1**	**2**	**7**	**6**	**6**	**2**
Weight of Each Bit	8^5	8^4	8^3	8^2	8^1	8^0
Weighted Value	1 X 8^5	2 X 8^4	7 X 8^3	6 X 8^2	6 X 8^1	2 X 8^0
Solved Multiplication	32768	8192	3584	384	48	2

Sum of weight of all bits = 32768 + 8192 + 3584 + 384 + 48 + 2
= 44978

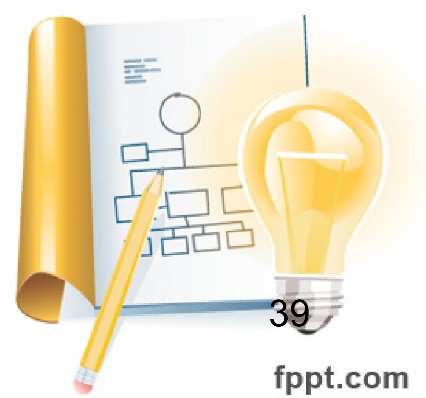


Conversion of Octal Fractions to Decimal Fractions

1. Determine the Decimal equivalent of (237.04)8

Octal Number	**2**	**3**	**7**	**0**	**4**
Weight of Each Bit	8^{2}	8^{1}	8^{0}	8^{-1}	8^{-2}
Weighted Value	2 X 8^{2}	3 X 8^{1}	7 X 8^{0}	0 X 8^{-1}	4 X 8^{-2}
Solved Multiplicatio n	128	24	7	0	0.0625

Sum of weight of all bits = 128 + 24 + 7 + 0 + 0.0625
= 159.0625



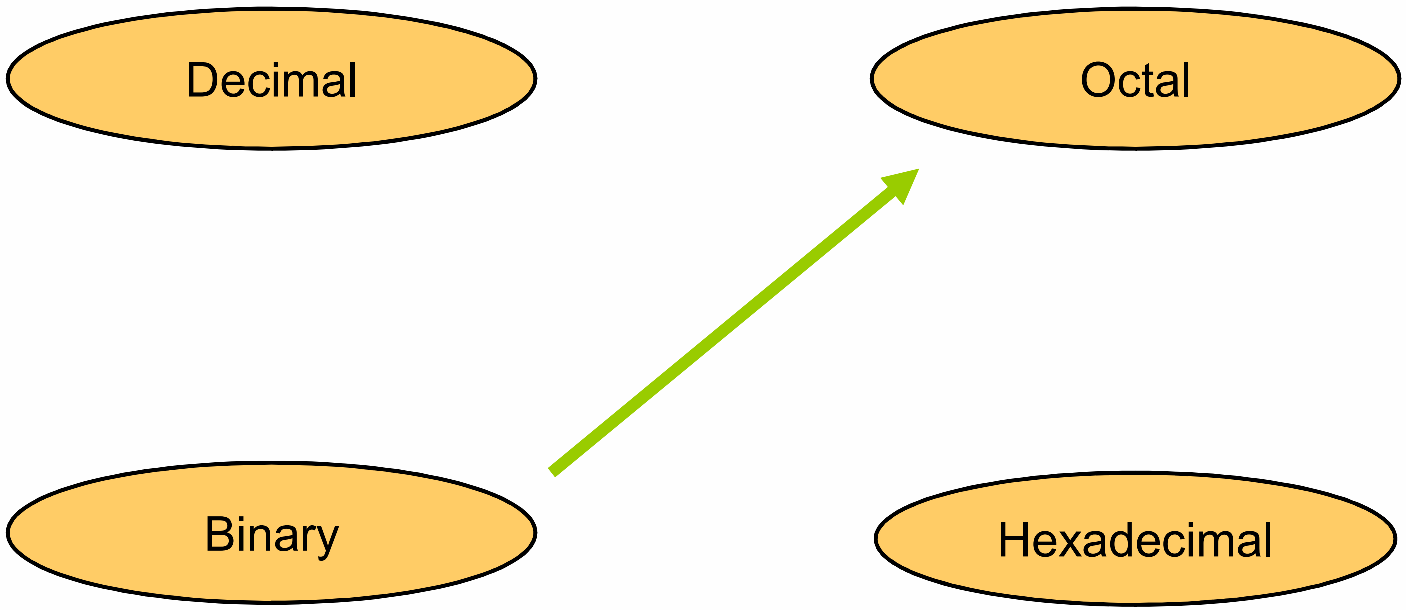
Conversion of Octal Fractions to Decimal Fractions

2. Determine the Decimal equivalent of (6732.032)8

Octal Number	**6**	**7**	**3**	**2**	**0**	**3**	**2**
Weight of Each Bit	8^{3}	8^{2}	8^{1}	8^{0}	8^{-1}	8^{-2}	8^{-3}
Weighted Value	6 X 8^{3}	7 X 8^{2}	3 X 8^{1}	2 X 8^{0}	0 X 8^{-1}	3 X 8^{-2}	2 X 8^{-3}
Solved Multiplication	3072	448	24	2	0	0.04687	0.00390

Sum of weight of all bits = 3072 + 448 + 24 + 2 +0+ 0.04687+0.00390
= 3546.05077

Binary to Octal



Binary to Octal

- Technique
  - Break the binary number into 3 sections starting from LSB to MSB (starting on right)
  - Convert to octal digits

Conversion of Binary to Octal

1. Determine the Octal equivalent of (1011010111)2

Binary Number	**001 (MSB)**	**011**	**010**	**111(LSB)**
Octal Number	1	3	2	7

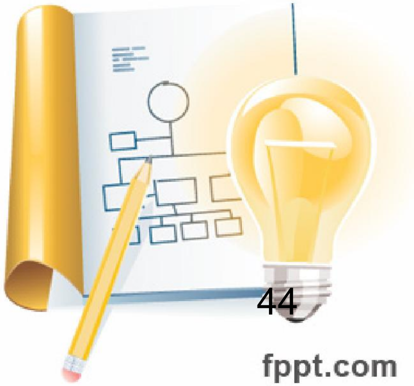
$$1011010111_2 = (1327)_8$$

Conversion of Binary to Octal

2. Determine the Octal equivalent of $(010111)_2$

Binary Number	010(MSB)	111 (LSB)
Octal Number	2	7

$$010111_2 = (27)_8$$

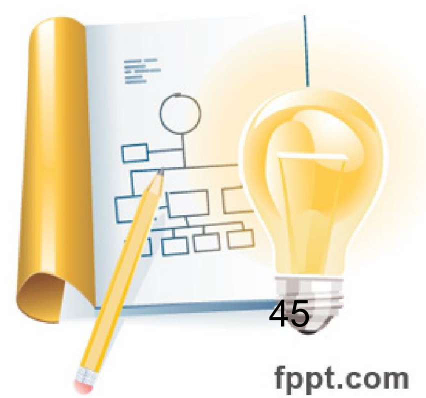


Conversion of Binary to Octal

3. Determine the Octal equivalent of $(1010111110110010)_2$

Binary Number	**001(MSB)**	**010**	**111**	**110**	**110**	**010(LSB)**
Octal Number	1	2	7	6	6	2

$$1010111110110010_2 = (127662)_8$$

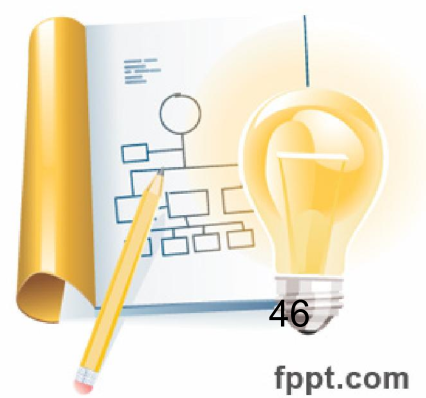


Conversion of Binary Fractions to Octal Fractions

1. Determine the Octal equivalent of $(0.110101)_2$

Binary Number	000	110(MSB)	101 (LSB)
Octal Number	0	6	5

$$0.110101_2 = (0.65)_8$$



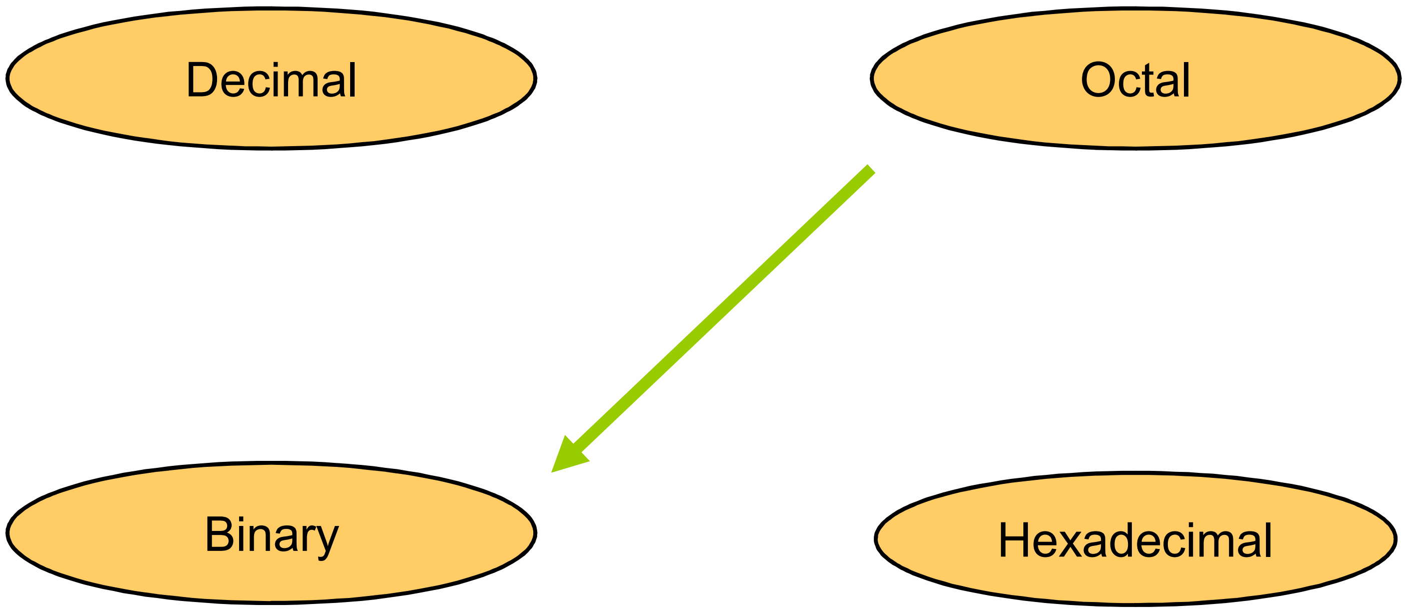
Conversion of Binary Fraction to Octal Fraction

2. Determine the Octal equivalent of (1100010.1110110)2

Binary Number	**001**	**100**	**010**	**111**	**011**	**000**
Octal Number	1	4	2	7	3	0

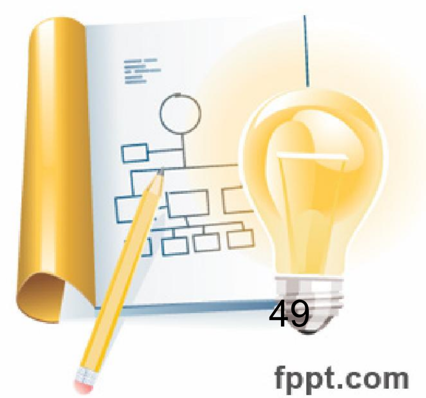
$$1100010.1110110_2 = (142.730)_8$$

Octal to Binary



Octal to Binary

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



Conversion of Octal to Binary

1. Determine the Binary equivalent of (705)8

Octal Number	**7**	**0**	**5**
Binary Coded Value	111	000	101

$$705_8 = (111000101)_2$$

Conversion of Octal to Binary

2. Determine the Binary equivalent of (231)8

Octal Number	2	3	1
Binary Coded Value	010	011	001

$$231_8 = (010011001)_2$$

Conversion of Octal to Binary

3. Determine the Binary equivalent of (453267)8

Octal Number	**4**	**5**	**3**	**2**	**6**	**7**
Binary Coded Value	100	101	011	010	110	111

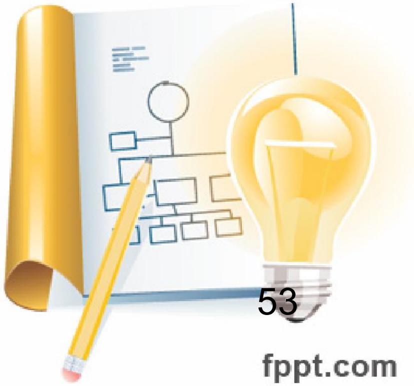
$$453267_8 = (100101011010110111)_2$$

Conversion of Octal Fractions to Binary Fractions

1. Determine the Binary equivalent of (2.335)8

Octal Number	2	3	3	5
Binary Coded Value	010	011	011	101

$$2.335_8 = (010.011011101)_2$$

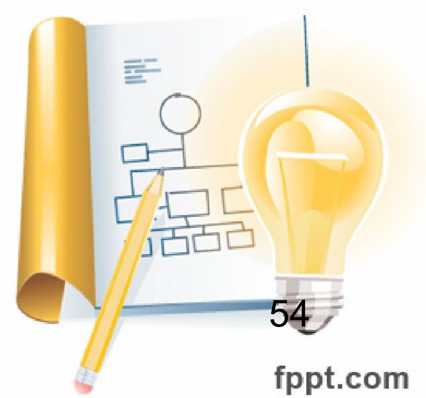


Conversion of Octal Fractions to Binary Fractions

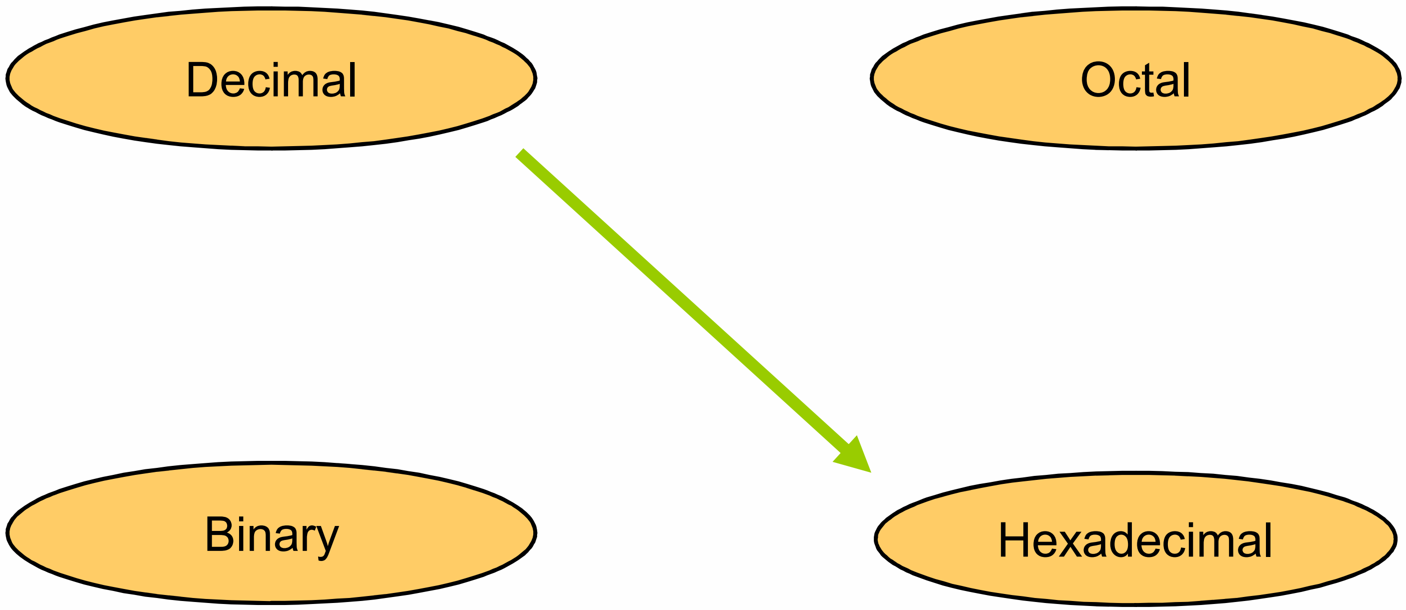
2. Determine the Binary equivalent of (5667.2411)8

Octal Number	5	6	6	7	2	4	1	1
Binary Coded Value	101	110	110	111	010	100	001	001

$$5667.2411_8 = (101110110111.010100001001)_2$$

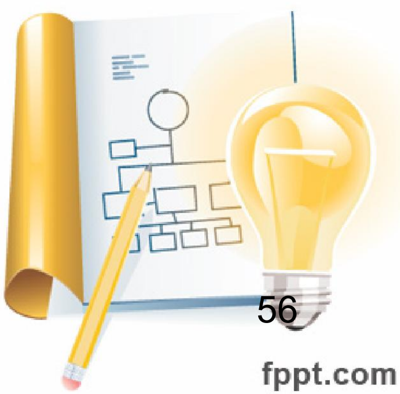


Decimal to Hexadecimal



Decimal to Hexadecimal

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



Conversion of Decimal to Hexadecimal

1. Determine the Hexadecimal equivalent of (1234)10

16	**1234**	**Remainder**
16	77	2
16	4	13
	0	4

Least Significant Bit (LSB)

Most Significant Bit (MSB)

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

Conversion of Decimal to Hexadecimal

2. Determine the Hexadecimal equivalent of (5112)10

16	5112	Remainder
16	319	8
16	19	15 = F
16	1	3
	0	1

Least Significant Bit (LSB)

Most Significant Bit (MSB)

$$5112_{10} = 13F8_{16}$$

Conversion of Decimal to Hexadecimal

3. Determine the Hexadecimal equivalent of (584666)10

16	584666	Remainder
16	36541	10=A
16	2283	13 = D
16	142	11=B
16	8	14=E
	0	8

Least Significant Bit (LSB)

Most Significant Bit (MSB)

$$584666_{10} = 8EBDA_{16}$$

Conversion of Decimal Fraction to Hexadecimal Fraction

1. Determine the Hexadecimal equivalent of (0.625)10

0.625	**X**	**16**	**= 10.000**	**10 = A**
0.000	X	16	= 0.000	0

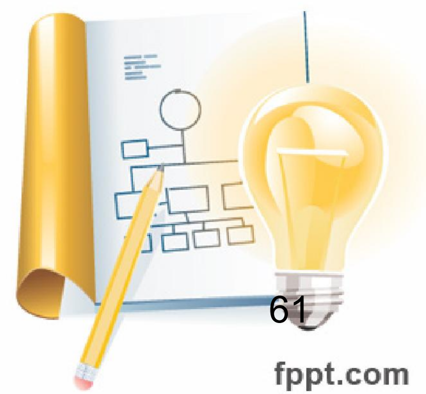
$$0.1325_{10} = 0.A0_{16}$$

Conversion of Decimal Fraction to Hexadecimal Fraction

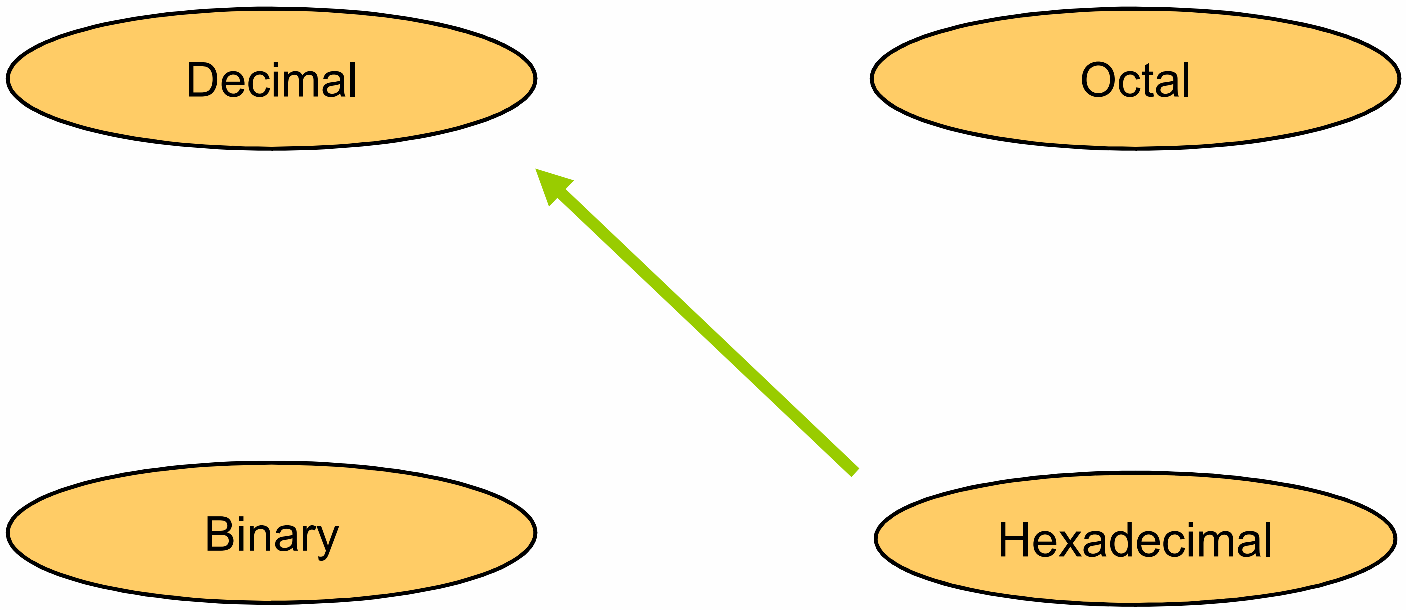
2. Determine the Hexadecimal equivalent of (0.2715)10

0.2715	**X**	**16**	**=**	**4.3440**	**4**
0.3440	X	16	=	5.5040	5
0.5040	X	16	=	8.0640	8
0.0640	X	16	=	1.0240	1
0.0240	X	16	=	0.3840	0
	.				
	.				
	∞				

$$0.2715_{10} = 0.45810_{16}$$



Hexadecimal to Decimal



Hexadecimal to Decimal

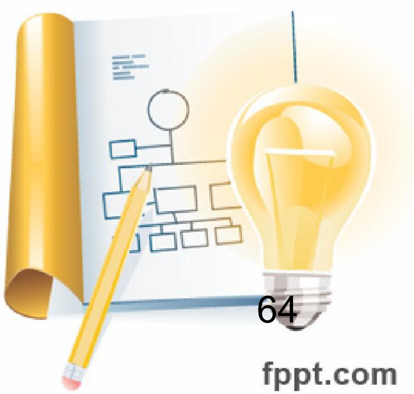
- Technique
  - Multiply each bit by 16^n, where n is the "weight" of the bit
  - The weight is the position of the bit, starting from 0 on the right
  - Add the results

Conversion of Hexadecimal to Decimal

1. Determine the Decimal equivalent of (B14)16

Hexadecimal Number	**B=11**	**1**	**4**
Weight of Each Bit	16^2	16^1	16^0
Weighted Value	11 X 16^2	1 X 16^1	4 X 16^0
Solved Multiplication	2816	16	4

Sum of weight of all bits = 2816 + 16 + 4
= 2836



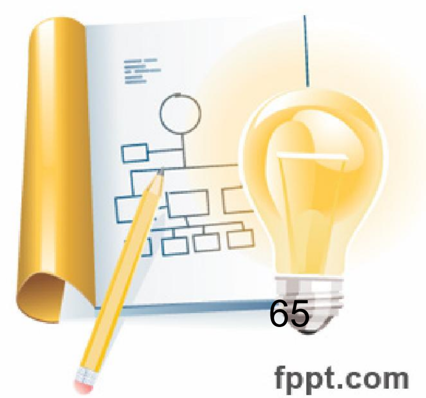
Conversion of Hexadecimal to Decimal

2. Determine the Decimal equivalent of (ABC)16

Hexadecimal Number	**A**	**B**	**C**
Weight of Each Bit	16^2	16^1	16^0
Weighted Value	10 X 16^2	11 X 16^1	12 X 16^0
Solved Multiplication	2560	176	12

Sum of weight of all bits = 2560+176+12

= 2748



Conversion of Hexadecimal to Decimal

3. Determine the Decimal equivalent of (8AFE2B)16

Hexadecimal Number	**8**	**A**	**F**	**E**	**2**	**B**
Weight of Each Bit	16^5	16^4	16^3	16^2	16^1	16^0
Weighted Value	8 X 16^5	10 X 16^4	15 X 16^3	14 X 16^2	2 X 16^1	11 X 16^0
Solved Multiplication	8388608	655360	61440	3584	32	11

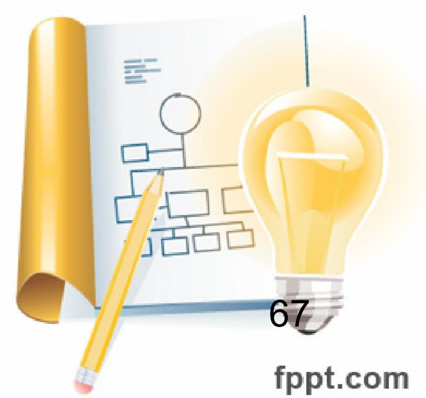
Sum of weight of all bits = 8388608+655360+61440+3584+32+11
= 9109035

Conversion of Hexadecimal Fractions to Decimal Fractions

1. Determine the Decimal equivalent of (A.23)16

Octal Number	**A**	**2**	**3**
Weight of Each Bit	16^0	16^{-1}	16^{-2}
Weighted Value	10 X 1	2 X 16^{-1}	3 X 16^{-2}
Solved Multiplication	10	0.125	0.01171875

Sum of weight of all bits = 10 + 0.125 + 0.01171875
= 10.13671875



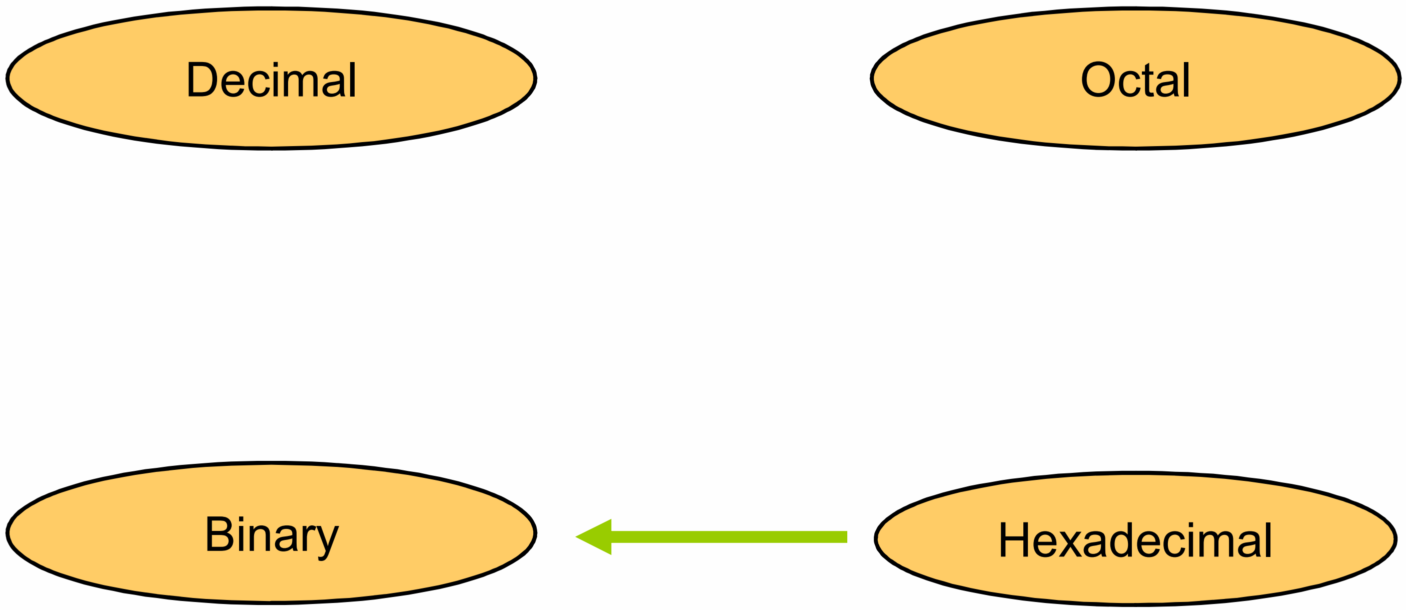
Conversion of Hexadecimal Fractions to Decimal Fractions

2. Determine the Decimal equivalent of (45C.8BE3)16

Octal Number	**4**	**5**	**C=12**	**8**	**B=11**	**E=14**	**3**
Weight of Each Bit	16^2	16^1	16^0	16^{-1}	16^{-2}	16^{-3}	16^{-4}
Weighted Value	4 X 256	5 X 16	12 X 1	8 X 16^{-1}	11 X 16^{-2}	14 X 16^{-3}	3 X 16^{-4}
Solved Multiplication	1024	80	12	0.5	0.0429687	0.0034179	0.0000457

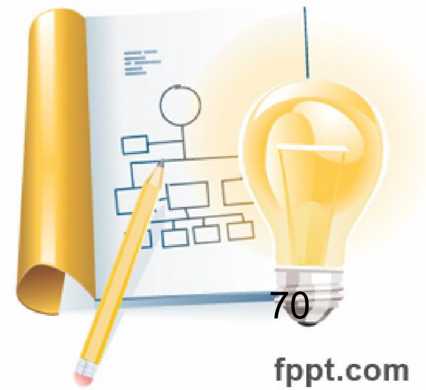
Sum of weight of all bits =
=1024 + 80 + 12 + 0.5+ 0.0429687+0.034179+0.0000457
= 1116.5464323

Hexadecimal to Binary



Hexadecimal to Binary

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

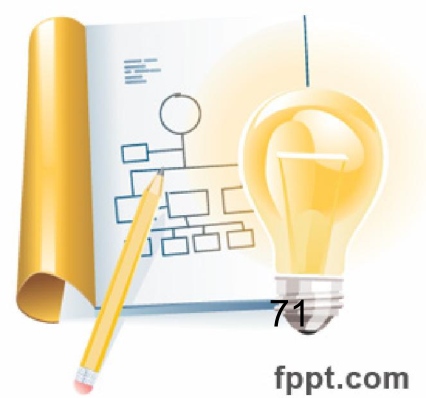


Conversion of Hexadecimal to Binary

1. Determine the Binary equivalent of (10AF)16

Hexadecimal Number	**1**	**0**	**A=10**	**F=15**
Binary Coded Value	0001	0000	1010	1111

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



Conversion of Hexadecimal to Binary

1. Determine the Binary equivalent of (5AF)16

Hexadecimal Number	**5**	**A=10**	**F=15**
Binary Coded Value	0101	1010	1111

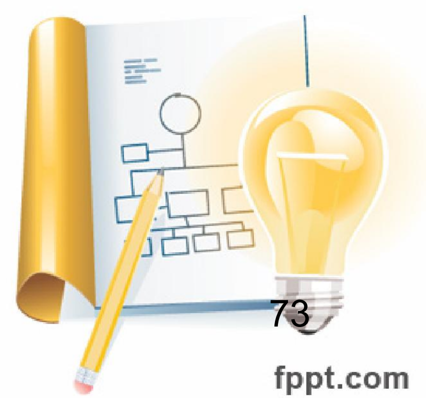
$$5AF_{16} = (010110101111)_2$$

Conversion of Hexadecimal to Binary

1. Determine the Binary equivalent of (86DB45C)16

Hexadecimal Number	**8**	**6**	**D=13**	**B=11**	**4**	**5**	**C=12**
Binary Coded Value	1000	0110	1101	1011	0100	0101	1100

$86DB45C_{16} = (1000011011011011010001011100)_2$



Conversion of Hexadecimal Fractions to Binary Fractions

1. Determine the Binary equivalent of (2B.6C)16

Hexadecimal Number	**2**	**B=11**	**6**	**C=12**
Binary Coded Value	0010	1011	0110	1100

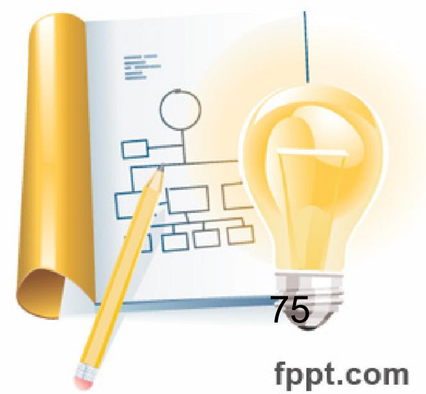
$$2B.6C_{16} = (00101011.01101100)_2$$

Conversion of Hexadecimal Fractions to Binary Fractions

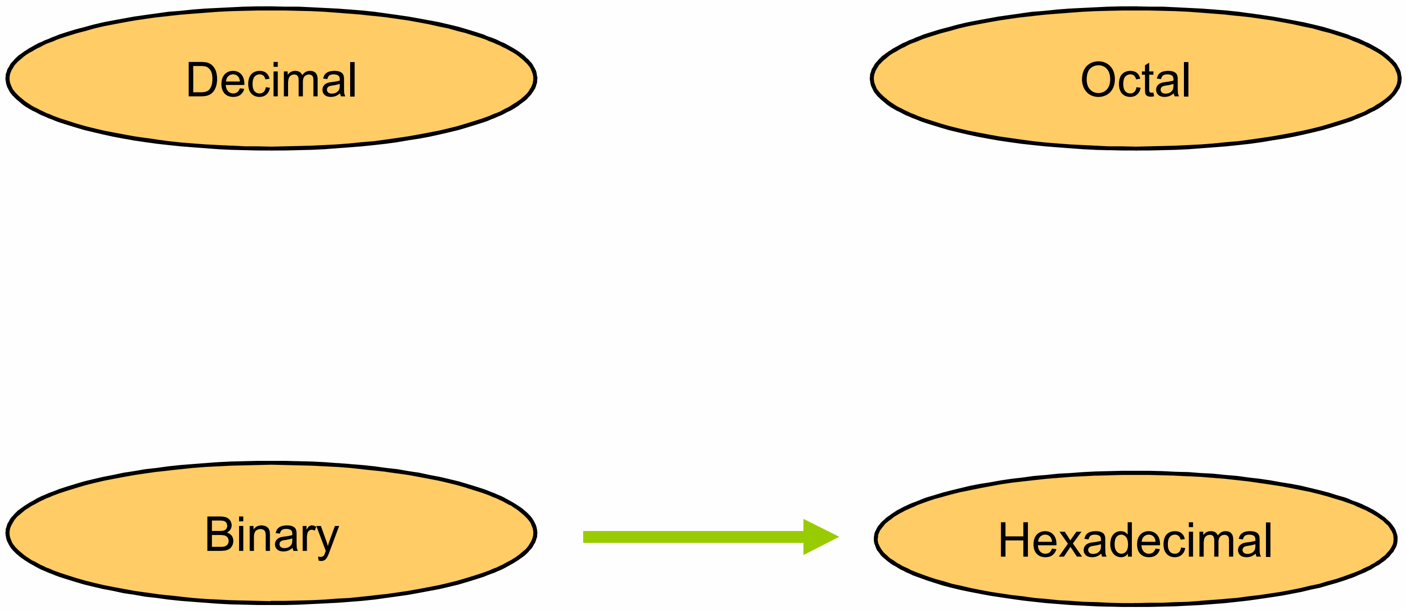
2. Determine the Binary equivalent of (576E.34DF)16

Hexadecimal Number	**5**	**7**	**6**	**E=14**	**3**	**4**	**D=13**	**F=15**
Binary Coded Value	0101	0111	0110	1110	0011	0100	1101	1111

$$576E.34DF_{16} = (0101011101101110.0011010011011111)_2$$

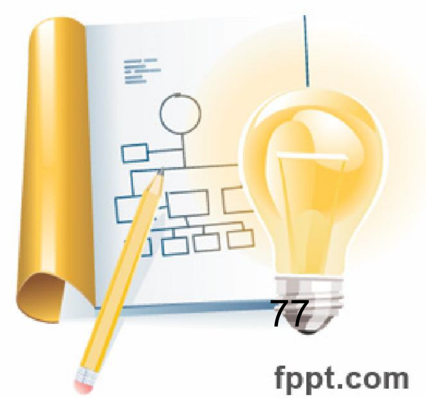


Binary to Hexadecimal



Binary to Hexadecimal

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

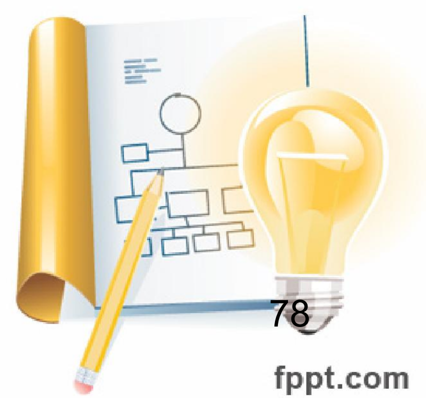


Conversion of Binary to Hexadecimal

1. Determine the Hexadecimal equivalent of $(1010111011)2$

Binary Number	0010	1011	1011
Decimal Number	2	11	11
Hexadecimal Number	2	B	B

$$1010111011_2 = (2BB)_{16}$$



Conversion of Binary to Hexadecimal

2. Determine the Hexadecimal equivalent of (11001011)2

Binary Number	1100	1011
Decimal Number	12	11
Hexadecimal Number	C	B

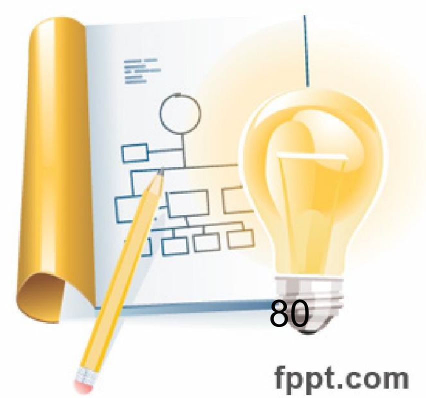
$$11001011_2 = (CB)_{16}$$

Conversion of Binary to Hexadecimal

3. Determine the Hexadecimal equivalent of $(10101111001101 1001)_2$

Binary Number	0010	1011	1100	1101	1001
Decimal Number	2	11	12	13	9
Hexadecimal Number	2	B	C	D	9

$$11001011_2 = (2BCD9)_{16}$$

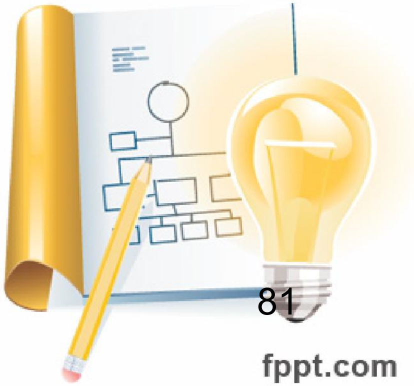


Conversion of Binary Fraction to Hexadecimal Fraction

1. Determine the Hexadecimal equivalent of (0.11101000)2

Binary Number	**0000**	**1110**	**1000**
Decimal Number	0	14	8
Hexadecimal Number	0	E	8

$$0.11101000_2 = (0.E8)_{16}$$



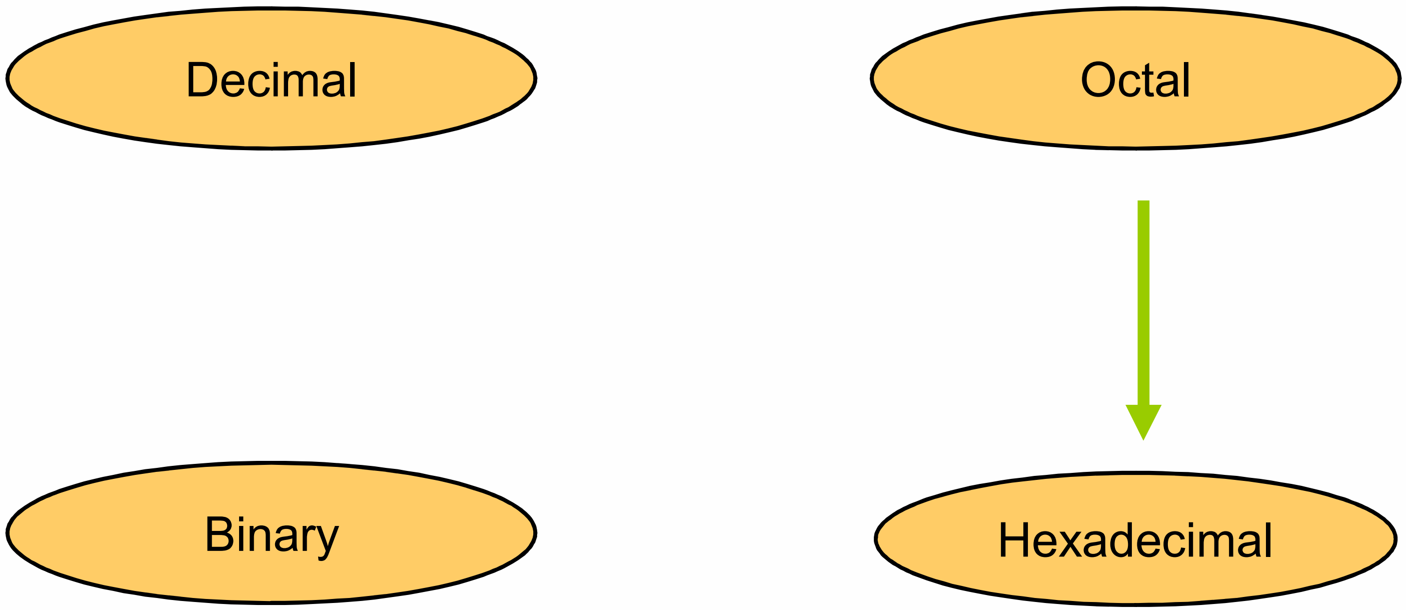
Conversion of Binary Fraction to Hexadecimal Fraction

2. Determine the Hexadecimal equivalent of $(1100001.101011110011)_2$

Binary Number	0110	0001	1010	1111	0011
Decimal Number	6	1	10	15	3
Hexadecimal Number	6	1	A	F	3

$$1100001.101011110011_2 = (61.AF3)_{16}$$

Octal to Hexadecimal



Octal to Hexadecimal

- Technique
  - Use binary as an intermediary

Conversion of Octal to Hexadecimal

1. Determine the Hexadecimal equivalent of (2327)8

Octal Number	2	3	2	7
Binary Coded Value	010	011	010	111

$$2327_8 = (\mathbf{0100\ 1101\ 0111})_2$$

4 13 7

$$2327_8 = (4D7)_{16}$$

Conversion of Octal to Hexadecimal

2. Determine the Hexadecimal equivalent of $(1076)8$

Octal Number	**1**	**0**	**7**	**6**
Binary Coded Value	001	000	111	110

$$1076_8 = (\mathbf{0010\ \ 0011\ \ 1110})_2$$

2 3 14

$$1076_8 = (23E)_{16}$$

Conversion of Octal Fraction to Hexadecimal Fraction

1. Determine the Hexadecimal equivalent of (31.57)8

Octal Number	**3**	**1**	**5**	**7**
Binary Coded Value	011	001	101	111

$$31.57_8 = (\mathbf{011001.101111})_2$$

$$= (\mathbf{0001\ 1001.1011\ 1100})$$

$$1 \quad 9 \quad . \quad 11 \quad 12$$

$$31.57_8 = (19.BC)_{16}$$

Conversion of Octal Fraction to Hexadecimal Fraction

2. Determine the Hexadecimal equivalent of (76.665)8

Octal Number	7	6	6	6	5
Binary Coded Value	111	110	110	110	101

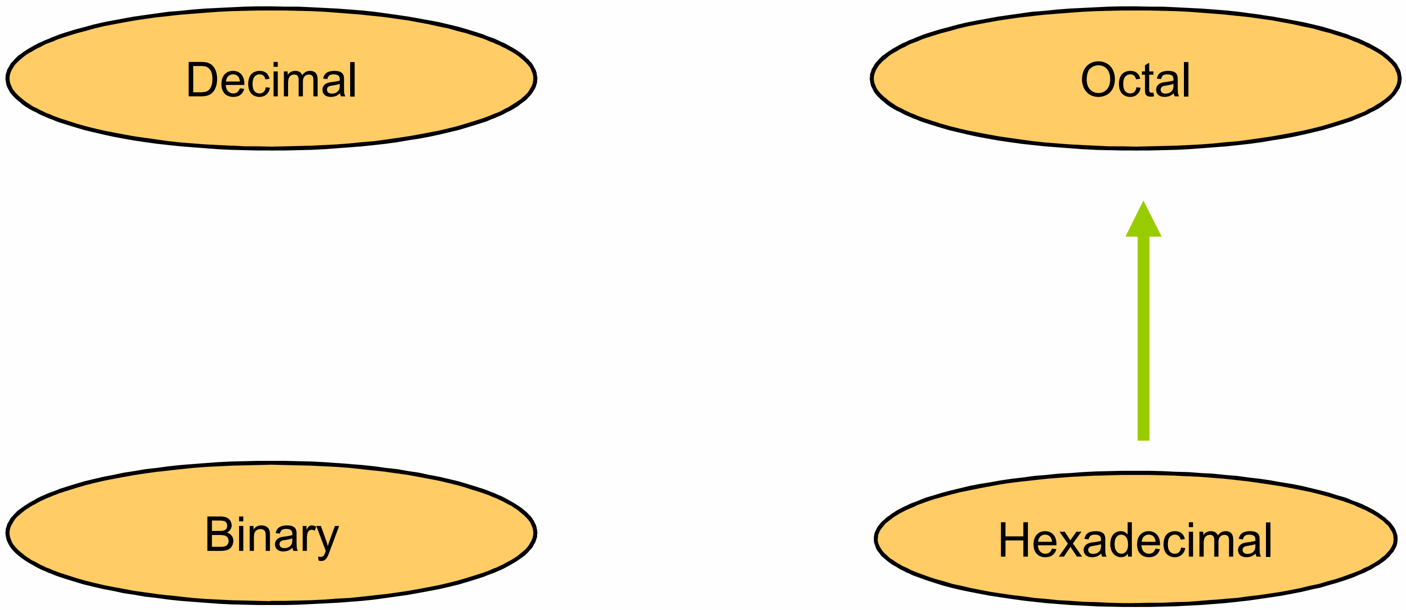
$$76.665_8 = (\mathbf{111110.110110101})_2$$

$$= (\mathbf{0011\ 1110\ .\ 1101\ 1010\ 1000})$$

$$3 \quad 14 \quad . \quad 13 \quad 10 \quad 8$$

$$76.665_8 = (3E.DA8)_{16}$$

Hexadecimal to Octal



Hexadecimal to Octal

- Technique
  - Use binary as an intermediary

Conversion of Hexadecimal to Octal

1. Determine the Octal equivalent of (2B6)16

Hexa decimal Number	**2**	**B=11**	**6**
Binary Coded Value	0010	1011	0110

$$2B6_{16} = (\mathbf{001010110110})_2$$

$$= (\mathbf{001\ \ 010\ \ 110\ \ 110})$$

$$1 \quad 2 \quad 6 \quad 6$$

$$2B6_{16} = (1266)_8$$

Conversion of Hexadecimal to Octal

2. Determine the Octal equivalent of (1F0C)16

Hexa decimal Number	1	F=15	0	C=12
Binary Coded Value	0001	1111	0000	1100

$$1F0C_{16} = (\mathbf{0001111100001100})_2$$

$$= (\mathbf{000\ 001\ 111\ 100\ 001\ 100})$$

0 1 7 4 1 4

$$1F0C_{16} = (017414)_8$$

Conversion of Hexadecimal to Octal

3. Determine the Octal equivalent of (5DE247)16

Hexa decimal Number	**5**	**D=13**	**E=14**	**2**	**4**	**7**
Binary Coded Value	0101	1101	1110	0010	0100	0111

$$5DE247_{16} = (\mathbf{010111011110001001000111})_2$$

$$= (\mathbf{010\ 111\ 011\ 110\ 001\ 001\ 000\ 111})$$

$$2 \quad 7 \quad 3 \quad 6 \quad 1 \quad 1 \quad 0 \quad 7$$

$$5DE247_{16} = (27361107)_8$$

Conversion of Hexadecimal Fractions to Octal Fractions

1. Determine the Octal equivalent of $(4.3C)16$

Hexa decimal Number	**4**	**3**	**C=12**
Binary Coded Value	0100	0011	1100

$$4.3C_{16} = (\mathbf{0100.00111100})_2$$

$$= (\mathbf{000\ 100\ .\ 001\ 111\ 000})$$

$$0 \quad 4 \quad . \quad 1 \quad 7 \quad 0$$

$$4.3C_{16} = (04.170)_8$$

Conversion of Hexadecimal Fractions to Octal Fractions

2. Determine the Octal equivalent of (7B.64D)16

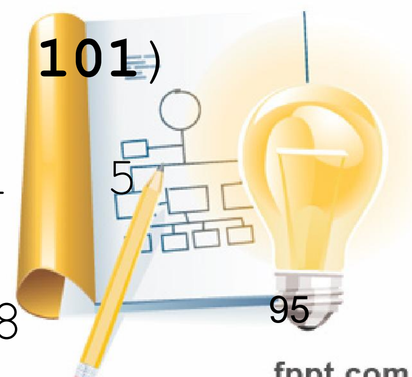
Hexa decimal Number	7	B=11	6	4	D=13
Binary Coded Value	0111	1011	0110	0100	1101

$$7B.64D_{16} = (\mathbf{01111011.011001001101})_2$$

$$= (\mathbf{001\ 111\ 011.011\ 001\ 001\ 101})$$

$$1 \quad 7 \quad 3\ .\ 3 \quad 1 \quad 1 \quad 5$$

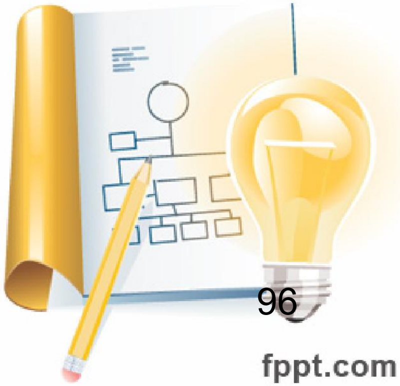
$$7B.64D_{16} = (173.3115)_8$$



Exercise – Convert ...

Decimal	Binary	Octal	Hexa-decimal
33			
	1110101		
		703	
			1AF

Don't use a calculator!



Exercise – Convert …

Answer

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

Exercise – Convert ...

Decimal	Binary	Octal	Hexa-decimal
29.8			
	101.1101		
		3.07	
			C.82

Don’t use a calculator!

Exercise – Convert …

Answer

Decimal	Binary	Octal	Hexa-decimal
29.8	11101.110011…	35.63…	1D.CC…
5.8125	101.1101	5.64	5.D
3.109375	11.000111	3.07	3.1C
12.5078125	1100.10000010	14.404	C.82

Arithmetic System

Binary Addition

- Addition of Binary Numbers

INPUT		OUTPUT	
X	Y	SUM(S)	CARRY(C)
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

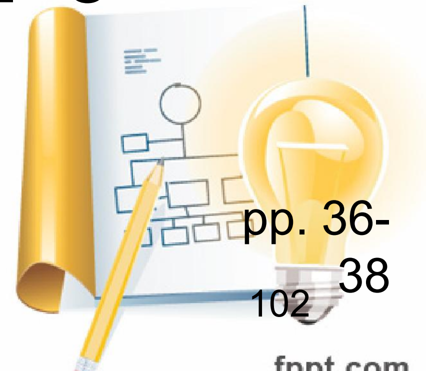
pp. 36-38

Binary Addition

- Add binary numbers 1111 and 1010

Binary **Decimal**

	Binary	Decimal
	1 1 1	
	1 1 1 1	1 5
+	1 0 1 0	+ 1 0
	1 1 0 0 1	2 5



pp. 36-38

Binary Addition

- Add binary numbers 110011, 10010, 1100 and 101

Binary	**Decimal**
1 1 1 1 1 1	**1**
1 1 0 0 1 1	5 1
1 0 0 1 0	1 8
1 1 0 0	1 2
+ 1 0 1	+ 5
1 0 1 0 1 1 0	8 6

Binary Addition

- Add binary numbers 11.10, 10.10

Binary

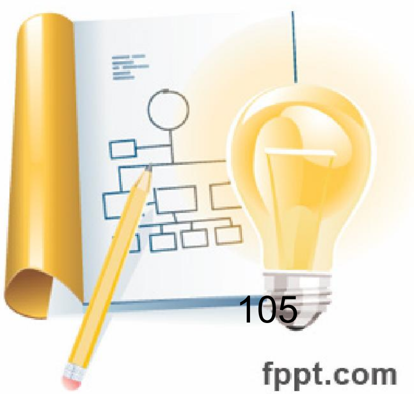
	Binary	Decimal
carry	1 **1 1**	**1**
	1 1 . 1 0	3 . 5
+	1 0 . 1 0	2 . 5
	1 1 0 . 0 0	6 . 0

Binary Addition

- Add binary numbers 11010.0100, 1001.01 ,001.11 and 10.1010

Binary

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



Binary Subtraction

- Subtraction of Binary Numbers

INPUT		OUTPUT	
X	Y	Difference(D)	Borrow(B)
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

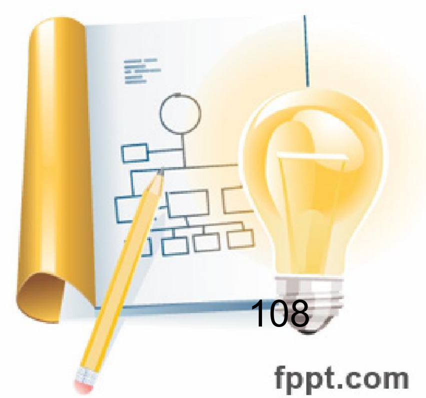
Example

	1				
	~~10~~	10	0	10	
~~1~~	~~0~~	~~0~~	~~1~~	~~0~~	1
0	0	1	0	1	1
0	1	1	0	1	0

Example

Find the binary difference of 1101 - 10110

		10	
1	~~1~~	~~0~~	1
1	0	1	1
0	0	1	0

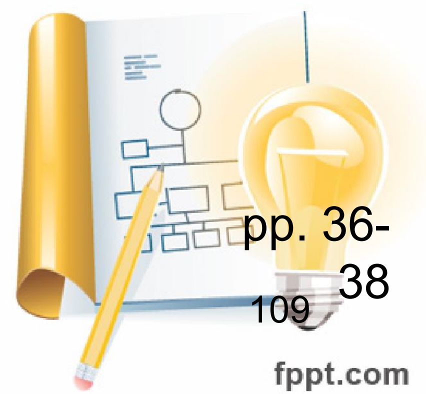


Example

Calculate the binary difference of

11100011 - 10101000

		10	1				
		~~0~~	~~10~~	10			
1	~~1~~	~~1~~	~~0~~	~~0~~	0	1	1
1	0	1	0	1	0	0	0
0	0	1	1	1	0	1	1



pp. 36-38

Octal Addition

1. First, add the two digits of the unit column of the octal number in decimal.
2. During the process of addition, if the sum is less than or equal to 7, then it can be directly written as octal digit.
3. If the sum is greater than 7, then subtract 8 from that particular digit and carry 1 to the next digit position.

Octal Addition

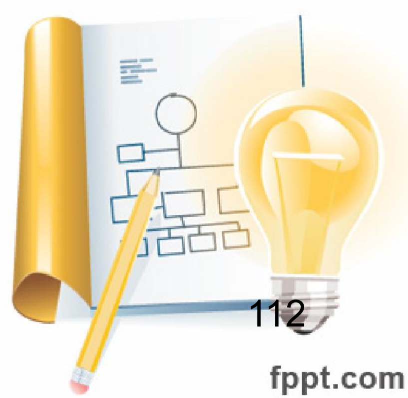
- Add the octal numbers 26 and 17.

	1	
	2	6
+	1	7
	4	13
-		8
	4	5

Octal Addition

- Add the octal numbers 5647 and 1425

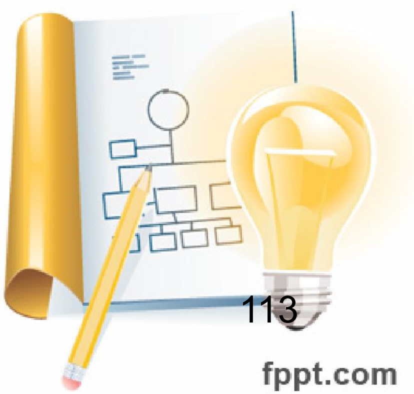
	1		1	
	5	6	4	7
+	1	4	2	5
	7	10	7	12
		-8		-8
	7	2	7	4



Octal Subtraction

Subtract the octal numbers 677 from 770

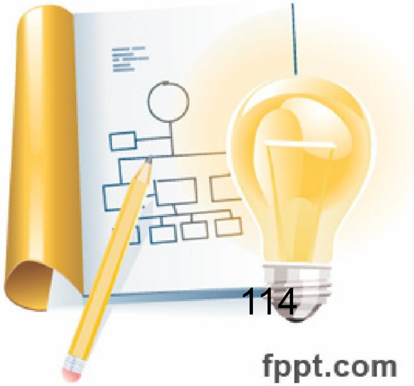
$$
\begin{array}{ccc}
 & 8+6=14 & \\
6 & \cancel{6} & 8 \\
\cancel{7} & \cancel{7} & \cancel{0} \\
6 & 7 & 7 \\
\hline
0 & 7 & 1
\end{array}
$$



Octal Subtraction

Subtract the octal numbers 2761 from 6357

5	8+2=10	8+5=13	
~~6~~	~~3~~	~~5~~	7
2	7	6	1
3	3	7	6



Hexadecimal Addition

1. First, add the two digits of the unit column of the octal number in decimal.
2. During the process of addition, if the sum is less than or equal to 15, then it can be directly written as a hexadecimal digit.
3. If the sum is greater than 15, then subtract 16 from that particular digit and carry 1 to the next digit position.

Hexadecimal Addition

- Add the hexadecimal numbers 76 and 45.

	7	6
+	4	5
	11	11
	-	-
	B	B

Hexadecimal Addition

- Add the hexadecimal numbers

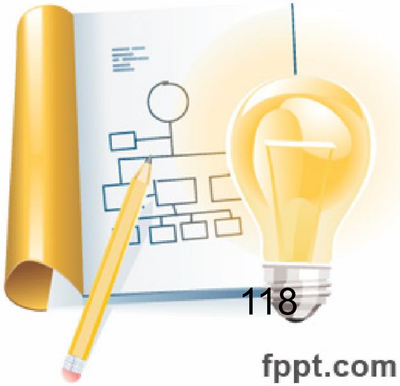
  A27E9 and 6FB43

		1	1	1		
		A	2	7	E	9
	+	6	F	B	4	3
1		17	18	19	18	12
		-16	-16	-16	-16	
1		1	2	3	2	12
1		1	2	3	2	C

Hexadecimal Subtraction

Subtract the hexadecimal numbers 75 from 527

4	16+2=18	
~~5~~	~~2~~	7
	7	5
4	11	2
4	B	2



Hexadecimal Subtraction

Subtract the hexadecimal numbers 1F65 from 7E2CA

	13	16+2=18		
7	~~E~~	~~2~~	C	A
	1	F	6	5
7	12	3	6	5
7	C	3	6	5

Binary Addition: Example

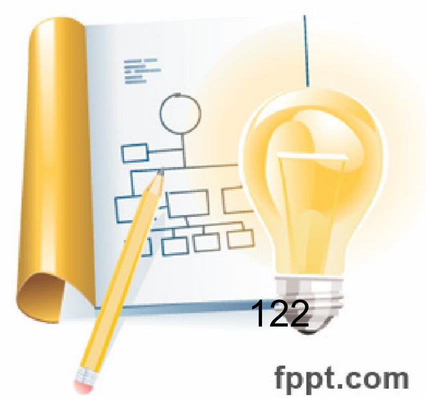
- 10001 + 11101 = ?
- 101101 + 11001 =?
- 1011001 + 111010 = ?
- 0011010 + 0001100 = ?

Binary Addition: Example

- 10001 + 11101 = 101110
- 101101 + 11001 = 1000110
- 1011001 + 111010 = 10010011
- 0011010 + 0001100 = 0100110

Binary Subtraction: Example

- 1011011 – 10010 = ?
- 1010110 – 101010 =?
- 100010110 – 1111010 = ?
- 1110110 – 1010111 = ?



Binary Subtraction: Example

- 1011011 – 10010 = 1001001
- 1010110 – 101010 = 0101100
- 100010110 – 1111010 = 010011100
- 1110110 – 1010111 = 0011111

Octal Addition: Example

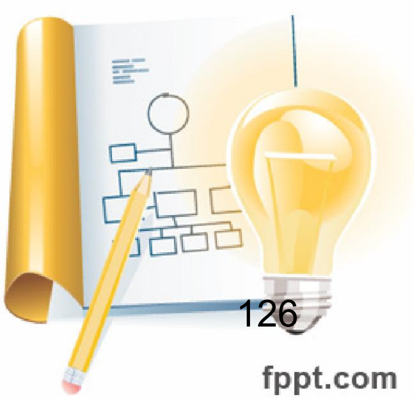
- 45667 + 2341 = ?
- 77542 + 16423 = ?
- 211345 + 456771 = ?

Octal Addition: Example

- 45667 + 2341 = 50230
- 77542 + 16423 = 116165
- 211345 + 456771 = 670336

Octal Subtraction: Example

- 76542 – 44367 = ?
- 123457 − 44663 = ?
- 456771 − 211345 = ?



Octal Subtraction: Example

- 76542 – 44367 = 32153
- 123457 − 44663 = 56574
- 456771 − 211345 = 245424

Hexadecimal Addition: Example

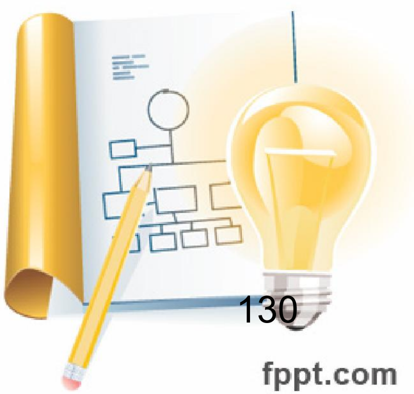
- A89EF + 347C = ?
- 2467 + 895A = ?
- 1B59A + 2E3FD = ?

Hexadecimal Addition: Example

- A89EF + 347C = ABE6B
- 2467 + 895A = ADC1
- 1B59A + 2E3FD = 49997

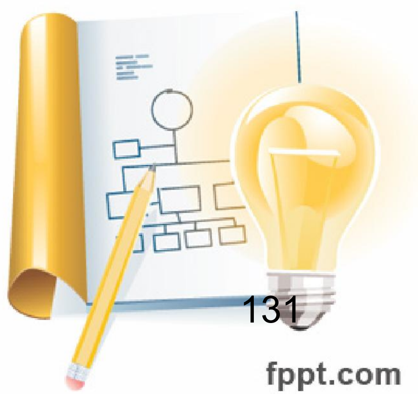
Hexadecimal Subtraction: Example

- E89B5 − 1FA27 = ?
- 6B432 − 59876 = ?
- 1B59A − 2E3D = ?
- ABCDEF − FEDCB = ?



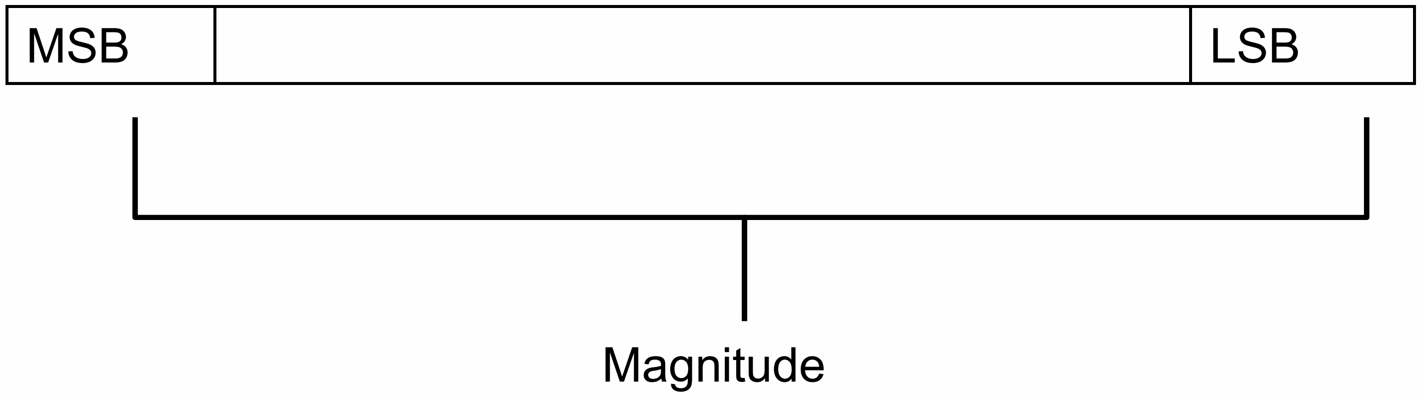
Hexadecimal Subtraction: Example

- E89B5 − 1FA27 = C8F8E
- 6B432 − 59876 = 11BBC
- 1B59A − 2E3D = 1875D
- ABCDEF − FEDCB = 9BE024

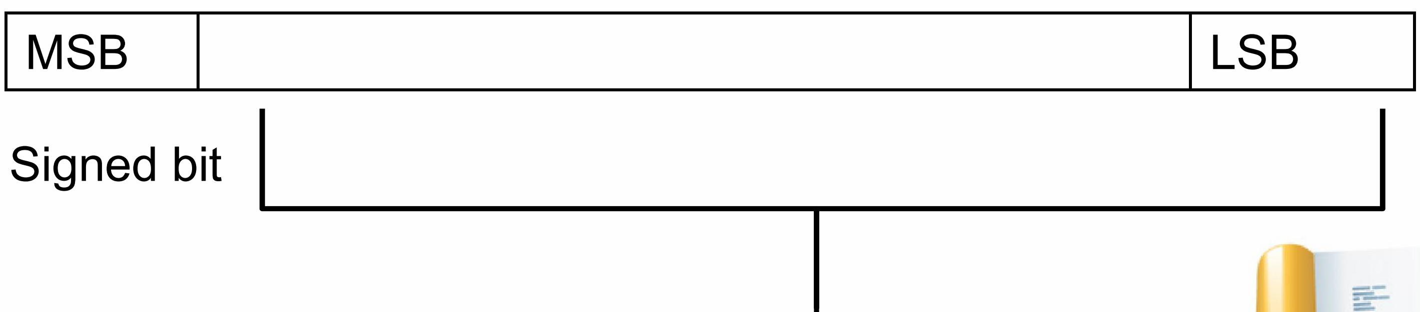


Signed And Unsigned Numbers

UnSigned Number



Signed Number



Magnitude

If MSB = 0 → Positive number

If MSB = 1 → Negative number

COMPLEMENT OF NUMBERS

Two types of complements for base R number system:

- R's complement and (R-1)'s complement

Example

- Decimal r= 10, 9's complement and 10’s complement
- Binary r=2, 1’s and 2’s complement
- Octal r=8, 7’s and 8’s complement

The R's Complement

Add 1 to the low-order digit of its (R-1)'s complement

SIGNED NUMBERS

Need to be able to represent both *positive* and *negative* numbers

- Following 3 representations

Signed magnitude representation
Signed 1's complement representation
Signed 2's complement representation

Example: Represent +9 and -9 in 7 bit-binary number

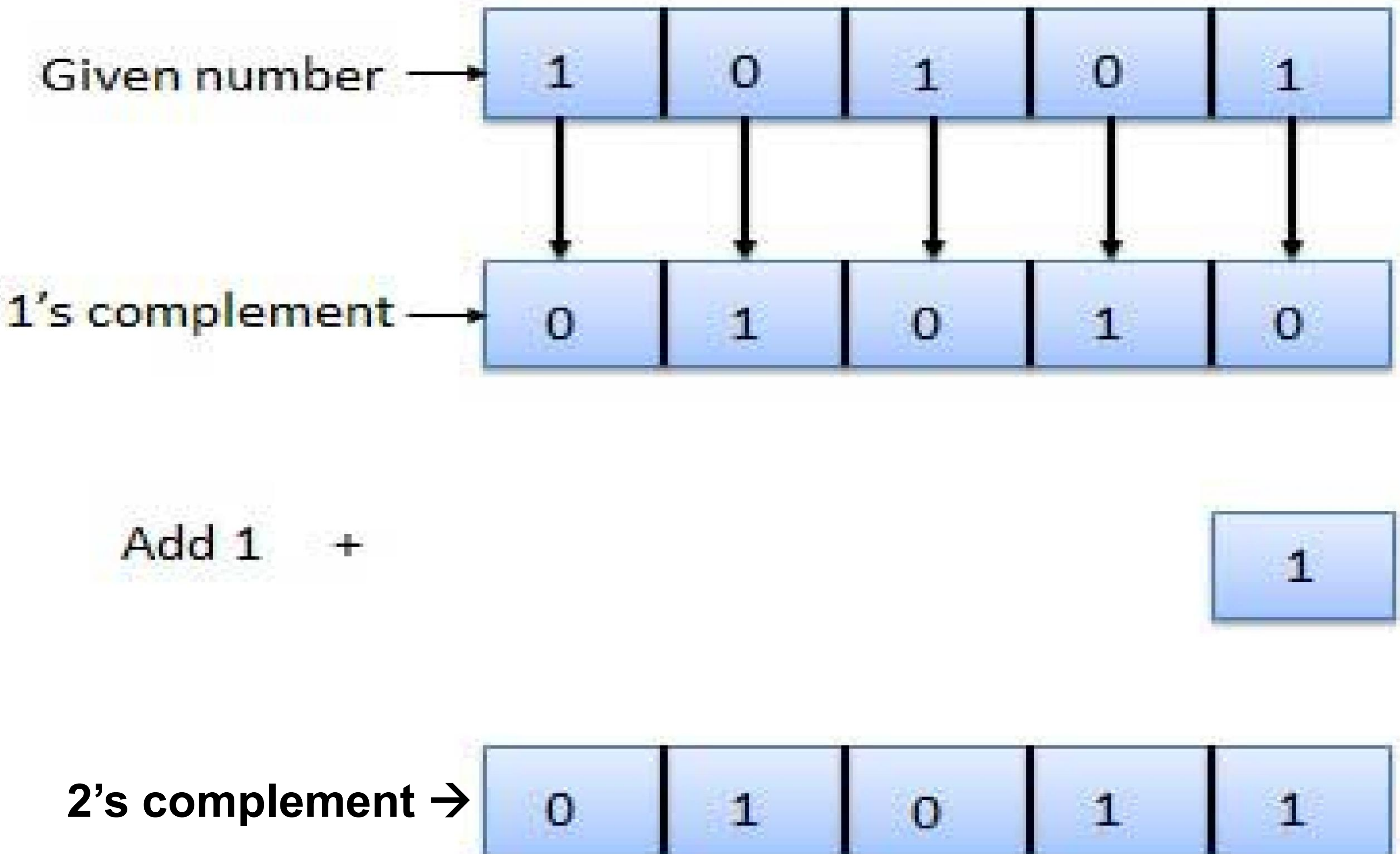
Only one way to represent +9 ==> 0 001001

Three different ways to represent -9:

In signed-magnitude: 1 001001

In signed-1's complement: 1 110110

In signed-2's complement: 1 110111


Given number
1 0 1 0 1
1's complement
0 1 0 1 0
Add 1 +
1
2's complement →
0 1 0 1 1

- (14) in 2’s complement form

Binary Number		1	1	1	0
1’s Complement		0	0	0	1
2’s Complement		0	0	1	0
With Sign Bit	1	0	0	1	0

+(12) in 2’s complement form

Binary Number		1	1	0	0
1’s Complement		0	0	1	1
2’s Complement		0	1	0	0
With Sign Bit	0	0	1	0	0

Addition-Subtraction of signed number using 2’s complement

1. Convert both numbers to equivalent binary form
2. Find the 2’s complement of subtrahend
3. Add this 2’s complement number to the minuend
4. If there is carry of 1, ignore it from the result to obtain the correct result.
5. If there is no carry, recomplement the result
    - attach the negative sign to the obtained result

Addition-Subtraction of signed number using 2's complement

1. Add $(27)_{10}$ and $(-11)_{10}$ using complementary representation for the negative value.

27 = 011011 and 11 = 001011

2's complement of (001011) = 1's complement of (001011) + 1

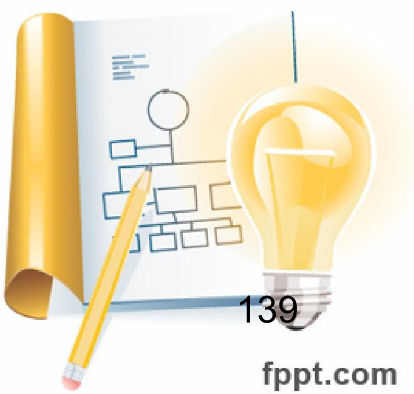
= 110100 + 1

= 110101

Add (011011) and (110101)

	1	1	1	1	1	
	0	1	1	0	1	1
+	1	1	0	1	0	1
1	0	1	0	0	0	0

Carry Hence, result is $(010000)_2$ or $(16)_{10}$



Addition-Subtraction of signed number using 2's complement

2. Subtract $(25)_{10}$ from $(42)_{10}$

25 = 011001 and 42 = 101010

2's complement of (011001) = 1's complement of (011001) + 1

= 100110 + 1

= 100111

Add (101010) and (100111)

			1	1	1		
		1	0	1	0	1	0
+		1	0	0	1	1	1
	1	0	1	0	0	0	1

Carry

Hence, Ignore carry and result is $(010001)_2$ or $(17)_{10}$

Addition-Subtraction of signed number using 2's complement

3. Subtract $(14)_{10}$ from $(46)_{10}$

14 = 00001110 and 46 = 00101110

2's complement of (00001110) = 1's complement of (00001110) + 1

= 11110001 + 1

= 11110010

Add (00101110) and (11110010)

			1		1		1		
		0	0	1	0	1	1	1	0
+		1	1	1	1	0	0	1	0
	1	0	0	1	0	0	0	0	0

Carry

Hence, Ignore carry and result is $(00100000)_2$ or $(32)_{10}$

Addition-Subtraction of signed number using 2's complement

4. Subtract 84 from 68 (68 – 84)

binary of 84 = 1010100

binary of 68 = 1000100

1's complement of 84 = 0101011

2's complement of 84 =

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

Now, add 1000100 + 0101100 = 1110000

There is no carry, so we will take 2's complement of result

Answer : - 0010000

Two's Complement Overflow Rules

- The rules for detecting overflow in a two's complement sum are simple:

1. If the sum of two positive numbers yields a negative result, the sum has overflowed.
2. If the sum of two negative numbers yields a positive result, the sum has underflowed.

Overflow

- Example: If we add two positive number 7 + 6 using 4-bit binary number, result should be +13

   0111 (7)

 + 0110 (6)

   1101

In signed notation, this is a result of -3, not +13

(because in 4 bit binary system, 4th bit represent sign bit and only 3 bits represent magnitude of the number.

Therefore, an overflow has occurred, where result would have more bits than the original numbers.

Underflow

- Example: If we add two negative numbers -29 and -13 using 8-bit binary number, result should be -42

29 = 00011101

1's of 29 = 11100010

2's of 29 =11100011

13 = 00001101

1's of 13 = 11110010

2's of 13 = 11110011

now, add 1 1 1 0 0 0 1 1

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

Result is 11010110

In signed notation, this is a result of +214 and not of -42.

Therefore, an underflow has occurred.

Cont…

- Overflow in two's complement occurs, not when a bit is carried out of the left column, but when there is a carry into the sign.
- A negative and positive added together cannot overflow, because the sum is between the addends.

- -39 + 92 = 53:

$\boxed{1}$	1		1	1				
	1	1	0	1	1	0	0	1
+	0	1	0	1	1	1	0	0
	0	0	1	1	0	1	0	1

Carryout without overflow. Sum is correct.

- 104 + 45 = 149:

	1	1		1				
	0	1	1	0	1	0	0	0
+	0	0	1	0	1	1	0	1
	1	0	0	1	0	1	0	1

Overflow, no carryout. Sum is not correct.

- 10 + -3 = 7:

$\boxed{1}$	1	1	1	1				
	0	0	0	0	1	0	1	0
+	1	1	1	1	1	1	0	1
	0	0	0	0	0	1	1	1

Carryout without overflow. Sum is correct.

Examples

- -75 + 59 = -16:

		1	1	1	1	1	1	
	1	0	1	1	0	1	0	1
+	0	0	1	1	1	0	1	1
	1	1	1	1	0	0	0	0

No overflow nor carryout.

- 127 + 1 = 128:

	1	1	1	1	1	1	1	
	0	1	1	1	1	1	1	1
+	0	0	0	0	0	0	0	1
	1	0	0	0	0	0	0	0

Overflow, no carryout. Sum is not correct.

- 44 + 45 = 89:

		1		1	1			
	0	0	1	0	1	1	0	0
+	0	0	1	0	1	1	0	1
	0	1	0	1	1	0	0	1

No overflow nor carryout.

- -103 + -69 = -172:

$\boxed{1}$		1	1	1		1	1	
	1	0	0	1	1	0	0	1
+	1	0	1	1	1	0	1	1
	0	1	0	1	0	1	0	0

Overflow, with incidental carryout. Sum is not currect.

- -1 + 1 = 0:

$\boxed{1}$	1	1	1	1	1	1	1	
	1	1	1	1	1	1	1	1
+	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0

Carryout without overflow. Sum is correct.

Basic Logic Gates

- Logic gate is an elementary building block of a digital circuit.
  – Which when combined with each other are able to perform complex logical and arithmetic operations.
- Two possible input

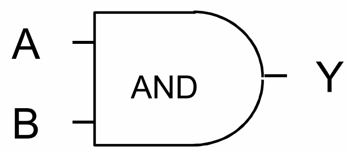
0 = 0v = False

1 = +5v = True

AND Function

Text Description ⇒ Output Y is TRUE if inputs A *AND* B are TRUE, else it is FALSE.

Logic Symbol ⇒



Truth Table ⇒

INPUTS		OUTPUT
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

AND Gate Truth Table

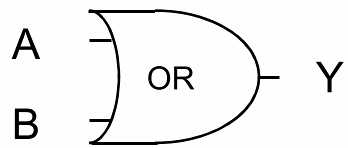
AND Symbol

Boolean Expression ⇒ $Y = A \times B = A \bullet B = AB$

OR Function

Text Description ⇒ Output Y is TRUE if input A *OR* B is TRUE, else it is FALSE.

Logic Symbol ⇒



Truth Table ⇒

INPUTS		OUTPUT
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

OR Gate Truth Table

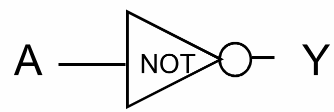
OR Symbol

Boolean Expression ⇒ $Y = A + B$

NOT Function (inverter)

Text Description ⇒ Output Y is TRUE if input A is FALSE, else it is FALSE. Y is the inverse of A.

Logic Symbol ⇒



Truth Table ⇒

INPUT A	OUTPUT Y
0	1
1	0

NOT Gate Truth Table

Boolean Expression ⇒ $Y = \overline{A}$

NOT Bar

Alternative Notation

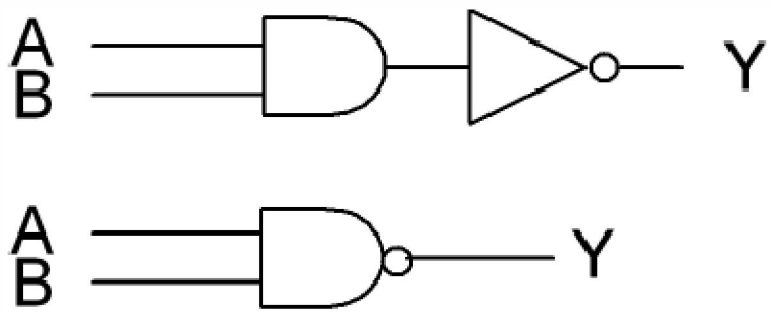
$Y = A'$

$Y = !A$

Combination of Logic Gates

NAND Function

- ➤The term NAND is formed by the combination of NOT-AND
  - implies an AND function with an inverted output.

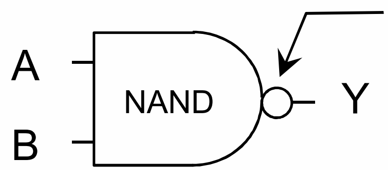


NAND Function

Text Description ⇒

Output Y is FALSE if inputs A ***AND*** B are TRUE, else it is TRUE.

Logic Symbol ⇒



A bubble is an inverter
This is an AND Gate with an inverted output

Truth Table ⇒

INPUTS		OUTPUT
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

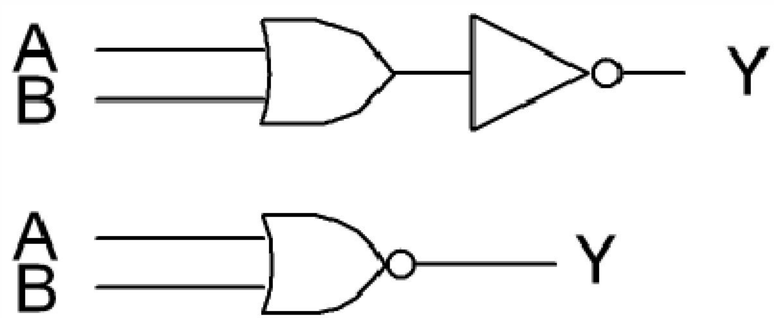
NAND Gate Truth Table

Boolean Expression ⇒

$$Y = \overline{A \times B} = \overline{AB}$$

NOR Function

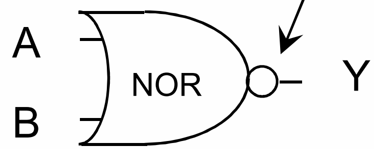
- The term NOR is formed by the combination of NOT-OR
  - implies an OR function with an inverted output.



NOR Function

Text Description ⇒ Output Y is FALSE if input A *OR* B is TRUE, else it is TRUE.

Logic Symbol ⇒



A bubble is an inverter.
This is an OR Gate with its output inverted.

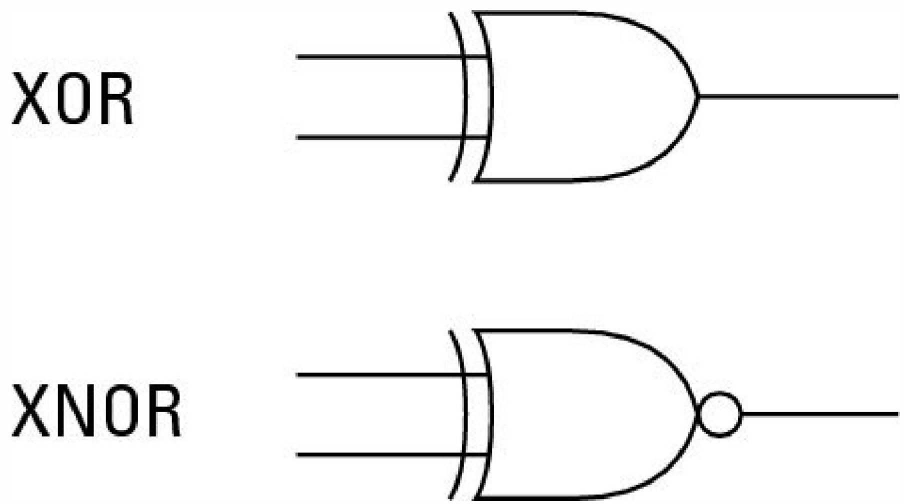
Truth Table ⇒

INPUTS		OUTPUT
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

NOR Gate Truth Table

Boolean Expression ⇒ $Y = \overline{A + B}$

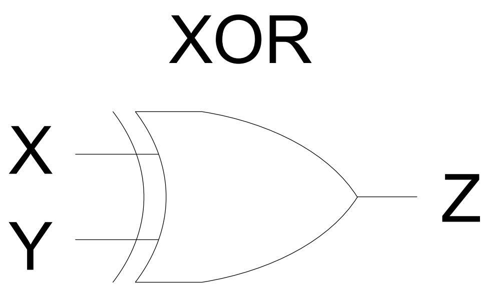
XOR and XNOR



➢XOR is an ‘inequality’ function → output is high(1) when the inputs are not equal to each other.

➢XNOR is an ‘equality’ function → output is high(1) when the inputs are equal to each other.

Exclusive-OR Gate

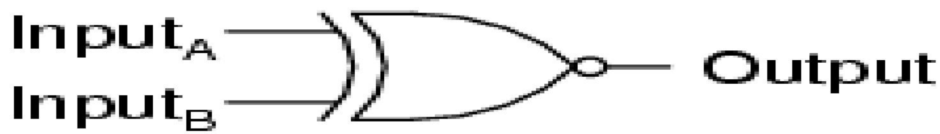


X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

$A \oplus B = A \cdot B' + A' \cdot B$

Exclusive-NOR Gate

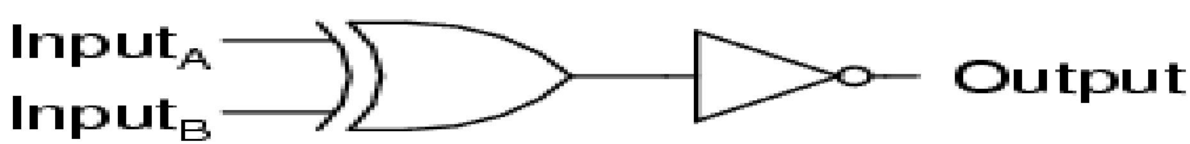
Exclusive-NOR gate



A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1

$A \odot B = A \bullet B + A' \bullet B'$

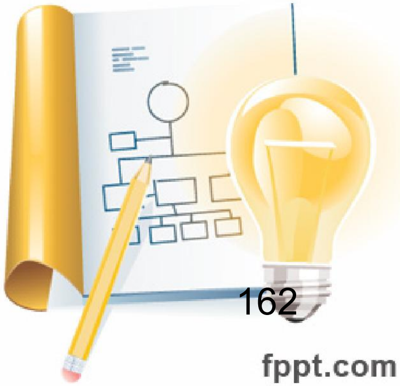
Equivalent gate circuit



Boolean Algebra

- Boolean algebra is the mathematics of digital systems.

- It is extensively used in designing the circuitry that is used in computers.

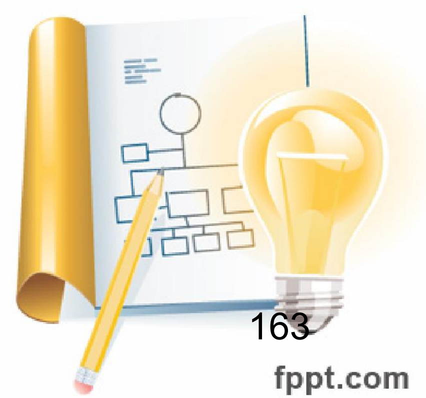


Boolean Operations

- The **complement** is denoted by a bar. It is defined by
- $\overline{0} = 1$ and $\overline{1} = 0$.

- The **Boolean sum**, denoted by + or by OR, has the following values:
- $1 + 1 = 1, \quad 1 + 0 = 1, \quad 0 + 1 = 1, \quad 0 + 0 = 0$

- The **Boolean product**, denoted by · or by AND, has the following values:
- $1 \cdot 1 = 1, \quad 1 \cdot 0 = 0, \quad 0 \cdot 1 = 0, \quad 0 \cdot 0 = 0$



Laws of Boolean Algebra

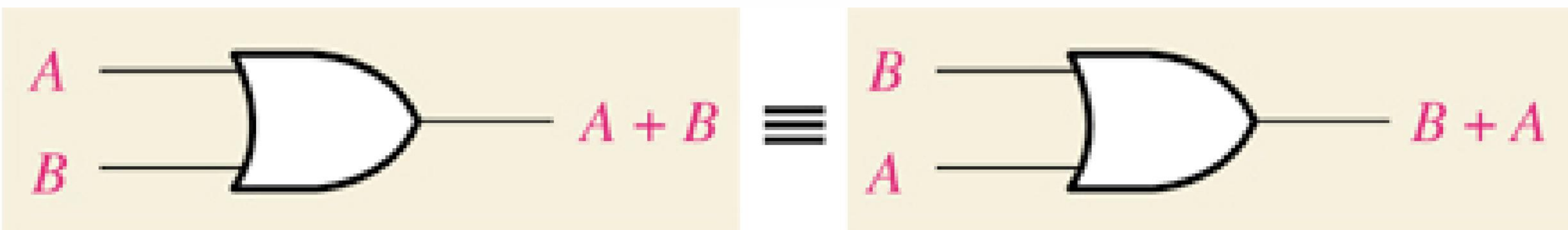
	AND Form	OR Form
Identify Law	$A.1 = A$	$A + 0 = A$
Zero and One Law	$A.0 = 0$	$A + 1=1$
Inverse Law	$A.\overline{A} = 0$	$A + \overline{A} =1$
Idempotent Law	$A.A = A$	$A + A = A$
Commutative Law	$A.B = B.A$	$A +B = B + A$
Associative Law	$A.(B.C) = (A.B).C$	$A + (B+C) = (A +B) + C$
Distributive Law	$A+(B.C)=(A+B).(A+C)$	$A.(B+C)= (A.B) + (A.C)$
Absorption Law	$A(A+B) = A$	$A + AB =A$ $A +A'B = A+B$
DeMorgan's Law	$\overline{(A.B)} = \overline{A} + \overline{B}$	$\overline{(A+B)} = \overline{A}.\overline{B}$
Double Complement Law	$\overline{\overline{X}}=X$	

Laws of Boolean Algebra

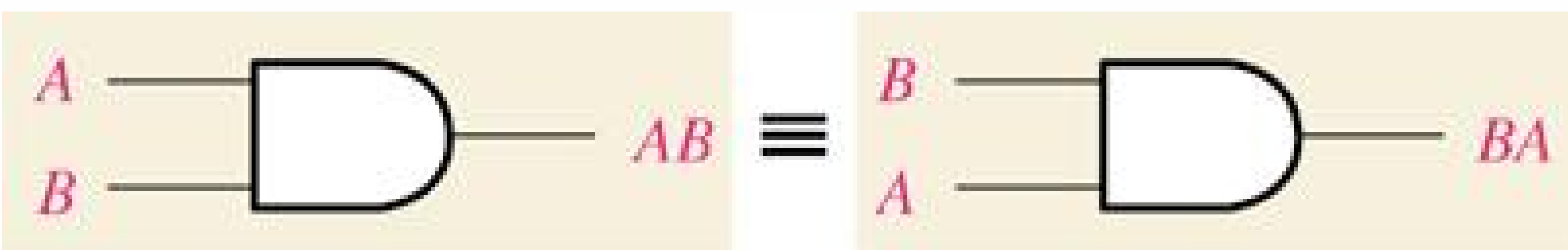
- Commutative Law

  the order of literals does not matter

  A + B = B + A



  A B = B A

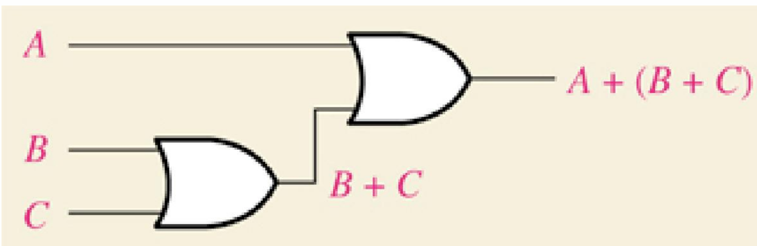


Laws of Boolean Algebra

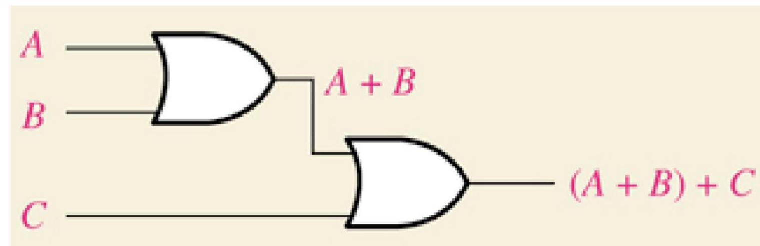
- Associative Law

  the grouping of literals does not matter

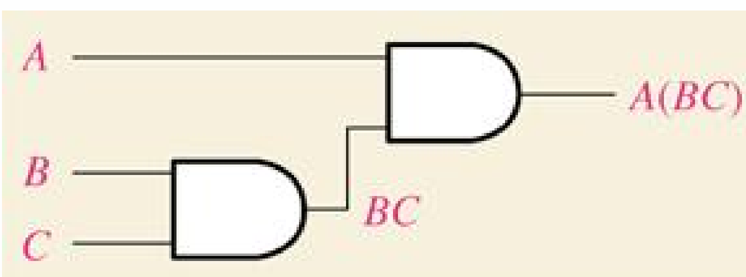
  A + (B + C) = (A + B) + C (=A+B+C)



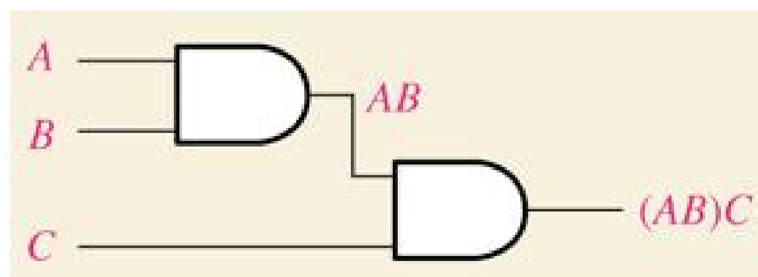
≡



  A(BC) = (AB)C (=ABC)

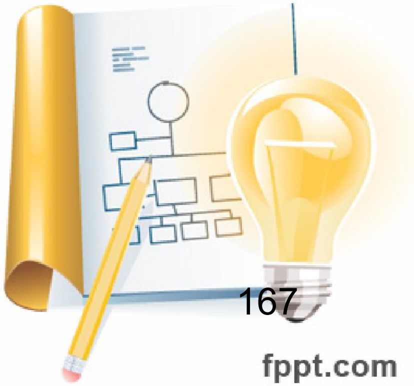


≡



Rules of Boolean Algebra

Rule Num ber	Boolean Expression
1	$A + 0 = A$
2	$A + 1 = 1$
3	$A \bullet 0 = 0$
4	$A \bullet 1 = A$
5	$A + A = A$
6	$A + A' = 1$
7	$A \bullet A = A$
8	$A \bullet A' = 0$
9	$A + AB = A$
10	$A + A'B = A + B$



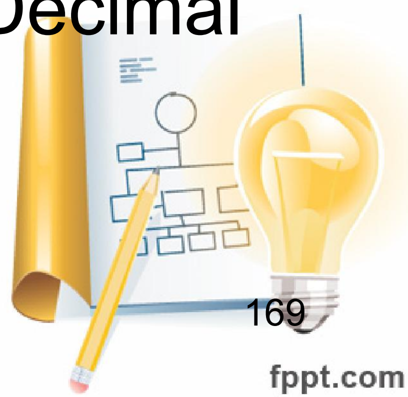
Why codes are used?

- Codes are used to represent the letters(A –Z, a-z) and special characters (such as +,-,*,$,&) in terms of 0’s and 1’s

- Every character can be represented by a combination of bits that is different from any other combination.

Coding Systems

- BCD - Binary Coded Decimal
- ASCII - American Standard Code for Information Interchange
- EBCDIC - Extended Binary Coded Decimal Interchange Code



Decimal and BCD

- The BCD is simply the 4 bit representation of the decimal digit.

- For multiple digit base 10 numbers, each symbol is represented by its BCD digit

- E.X.

5	3	1	9
0101	0011	0001	1001

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

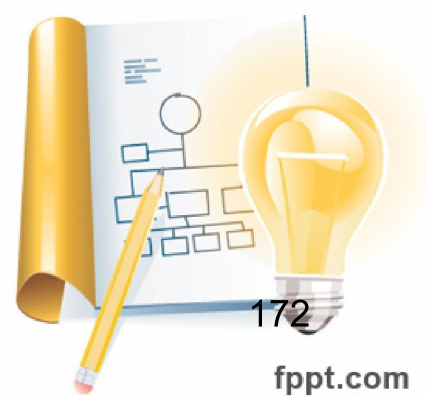
- BCD is fastest way to convert numbers from decimal to binary.
- However, it can represent only 16,(2^4) symbols.
- The later version of BCD used a 6-bit code, which allows representing a max. of 64 that is 2^6 symbols.
- However, it is also not sufficient for modern computers.

ASCII

- It stands for American Standard Code for Information Interchange

- It is widely used in micro computers, data transmission.
- Code was originally designed as 7 bit code

- Later on, IBM developed a new version of ASCII called as ASCII-8.
    - They use of all 8 bits providing 256 symbols

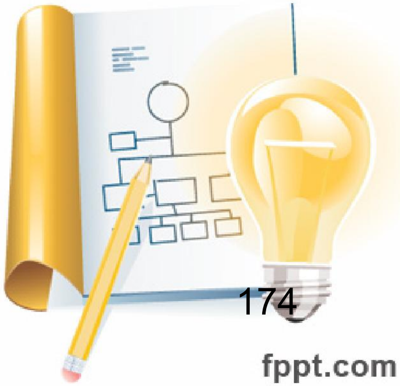


ASCII

0	<NUL>	32	<SPC>	64	@	96	`	128	Ä	160	†	192	¿	224	‡
1	<SOH>	33	!	65	A	97	a	129	Å	161	°	193	¡	225	·
2	<STX>	34	"	66	B	98	b	130	Ç	162	¢	194	¬	226	‚
3	<ETX>	35	#	67	C	99	c	131	É	163	£	195	√	227	„
4	<EOT>	36	$	68	D	100	d	132	Ñ	164	§	196	ƒ	228	‰
5	<ENQ>	37	%	69	E	101	e	133	Ö	165	•	197	≈	229	Â
6	<ACK>	38	&	70	F	102	f	134	Ü	166	¶	198	∆	230	Ê
7	<BEL>	39	'	71	G	103	g	135	á	167	ß	199	«	231	Á
8	<BS>	40	(	72	H	104	h	136	à	168	®	200	»	232	Ë
9	<TAB>	41	)	73	I	105	i	137	â	169	©	201	…	233	È
10	<LF>	42	*	74	J	106	j	138	ä	170	™	202		234	Í
11	<VT>	43	+	75	K	107	k	139	ã	171	´	203	À	235	Î
12	<FF>	44	,	76	L	108	l	140	å	172	¨	204	Ã	236	Ï
13	<CR>	45	-	77	M	109	m	141	ç	173	≠	205	Õ	237	Ì
14	<SO>	46	.	78	N	110	n	142	é	174	Æ	206	Œ	238	Ó
15	<SI>	47	/	79	O	111	o	143	è	175	Ø	207	œ	239	Ô
16	<DLE>	48	0	80	P	112	p	144	ê	176	∞	208	–	240	
17	<DC1>	49	1	81	Q	113	q	145	ë	177	±	209	—	241	Ò
18	<DC2>	50	2	82	R	114	r	146	í	178	≤	210	“	242	Ú
19	<DC3>	51	3	83	S	115	s	147	ì	179	≥	211	”	243	Û
20	<DC4>	52	4	84	T	116	t	148	î	180	¥	212	‘	244	Ù
21	<NAK>	53	5	85	U	117	u	149	ï	181	µ	213	’	245	ı
22	<SYN	54	6	86	V	118	v	150	ñ	182	∂	214	÷	246	ˆ
23	<ETB>	55	7	87	W	119	w	151	ó	183	∑	215	◊	247	˜
24	<CAN>	56	8	88	X	120	x	152	ò	184	∏	216	ÿ	248	¯
25	<EM>	57	9	89	Y	121	y	153	ô	185	π	217	Ÿ	249	˘
26	<SUB>	58	:	90	Z	122	z	154	ö	186	∫	218	⁄	250	˙
27	<ESC>	59	;	91	[	123	{	155	õ	187	ª	219	€	251	˚
28	<FS>	60	<	92	\	124	|	156	ú	188	º	220	‹	252	¸
29	<GS>	61	=	93	]	125	}	157	ù	189	Ω	221	›	253	˝
30	<RS>	62	>	94	^	126	~	158	û	190	æ	222	ﬁ	254	˛
31	<US>	63	?	95	_	127	<DEL>	159	ü	191	ø	223	ﬂ	255	ˇ

- Determine the binary code of ‘word’ in the ASCII form.

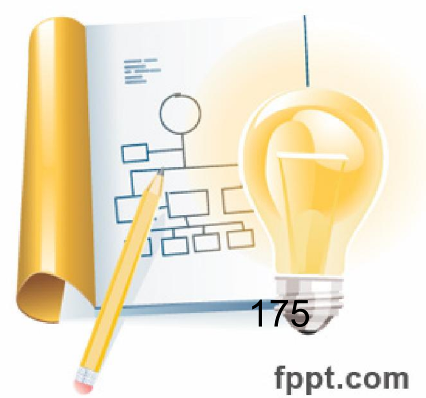
w	o	r	d
119	111	114	100
01110111	01101111	01110010	01100110



EBCDIC

- Extended Binary Coded Decimal Interchange Code uses 8 bits for each character.
  - Provides 256 different unique code

Character	Zone bits
A – I	1100
J – R	1101
S – Z	1110
0 - 9	1111
a – i	1000
j – r	1001
s – z	1010



ALPHABETIC CHARACTERS					
UPPERCASE			LOWERCASE		
PRINTS AS	EBCDIC IN BINARY	EBCDIC IN HEXADECIMAL	PRINTS AS	EBCDIC IN BINARY	EBCDIC IN HEXADECIMAL
	zzzz 8421			zzzz 8421	
A	1100 0001	C 1	a	1000 0001	8 1
B	1100 0010	C 2	b	1000 0010	8 2
C	1100 0011	C 3	c	1000 0011	8 3
D	1100 0100	C 4	d	1000 0100	8 4
E	1100 0101	C 5	e	1000 0101	8 5
F	1100 0110	C 6	f	1000 0110	8 6
G	1100 0111	C 7	g	1000 0111	8 7
H	1100 1000	C 8	h	1000 1000	8 8
I	1100 1001	C 9	i	1000 1001	8 9
J	1101 0001	D 1	j	1001 0001	9 1
K	1101 0010	D 2	k	1001 0010	9 2
L	1101 0011	D 3	l	1001 0011	9 3
M	1101 0100	D 4	m	1001 0100	9 4
N	1101 0101	D 5	n	1001 0101	9 5
O	1101 0110	D 6	o	1001 0110	9 6
P	1101 0111	D 7	p	1001 0111	9 7
Q	1101 1000	D 8	q	1001 1000	9 8
R	1101 1001	D 9	r	1001 1001	9 9
S	1110 0010	E 2	s	1010 0010	A 2
T	1110 0011	E 3	t	1010 0011	A 3
U	1110 0100	E 4	u	1010 0100	A 4
V	1110 0101	E 5	v	1010 0101	A 5
W	1110 0110	E 6	w	1010 0110	A 6
X	1110 0111	E 7	x	1010 0111	A 7
Y	1110 1000	E 8	y	1010 1000	A 8
Z	1110 1001	E 9	z	1010 1001	A 9
NUMERIC CHARACTERS					
0	1111 0000	F 0	5	1111 0101	F 5
1	1111 0001	F 1	6	1111 0110	F 6
2	1111 0010	F 2	7	1111 0111	F 7
3	1111 0011	F 3	8	1111 1000	F 8
4	1111 0100	F 4	9	1111 1001	F 9

EBCDIC vs. ASCII

Character	D	P	3
EBCDIC	1100 0100	1101 0111	1111 0011
ASCII	0100 0100	0101 0000	0011 0011