Unit I – Fundamental of Logic Design

- ✓**Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- ✓**Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓Subtraction using 1's complement and 2's complement
- ✓**Codes:** BCD, Gray Code, Excess-3, ASCII code
- ✓**BCD Arithmetic:** BCD Addition

Unit-I

Fundamental of Logic Design

Analog Vs Digital

Analog Signal	Digital Signal
• Continuous	• Discrete
• Infinite range of values	• Finite range of values (2)
• More exact values, but more difficult to work with	• Not as exact as analog, but easier to work with

Example:

A digital thermostat in a room displays a temperature of 72°. An analog thermometer measures the room temperature at 72.482°. The analog value is continuous and more accurate, but the digital value is more than adequate for the application and significantly easier to process electronically.

Example of Analog Vs Digital

Digital advantages:

Battery life

Programmability

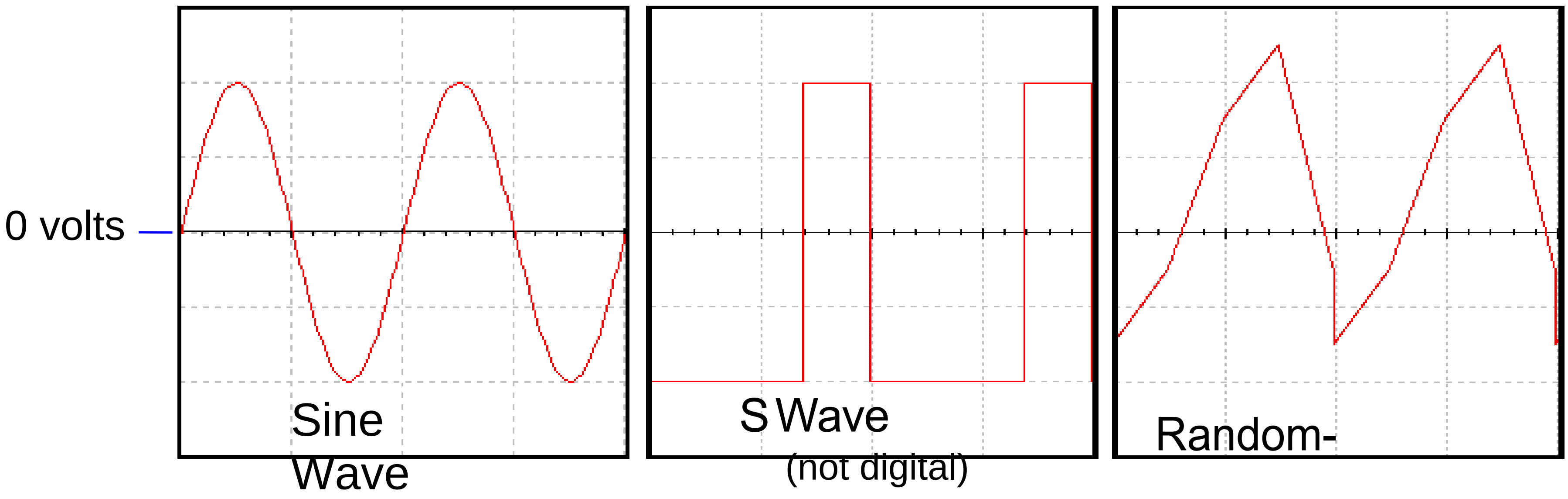
Accuracy

The World is

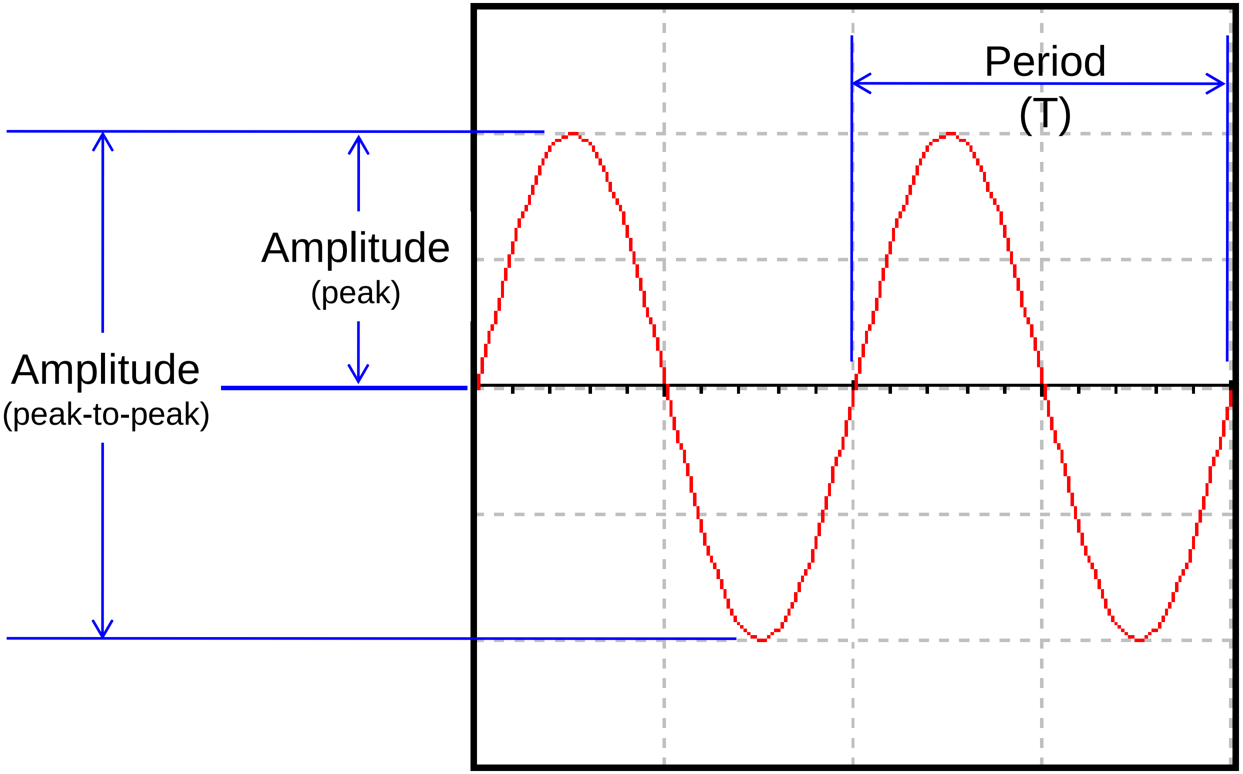
- ✓ The world we live in is analog.
- ✓ We are analog.
- ✓ Any inputs we can perceive are analog.
- ✓ For example,
  - – sounds are analog signals; they are continuous time and continuous value.
  - – Our ears listen to analog signals and we speak with analog signals.
  - – Images, pictures, and video are all analog at the source and our eyes are analog sensors.
  - – Measuring our heartbeat, tracking our activity, all requires processing analog sensor information.

Examples of Analog

- ✓ An analog signal can be any time-varying signal.
- ✓ Minimum and maximum values can be either positive or negative.
- ✓ They can be periodic (repeating) or non-periodic.
- ✓ Sine waves and square waves are two common analog signals.
- ✓ Note that this square wave is not a digital signal because its minimum value is negative.
- ✓ Video and Audio



Parts of Analog



Frequency:

$$F = \frac{1}{T}\,\text{Hz}$$

Pros and Cons Analog

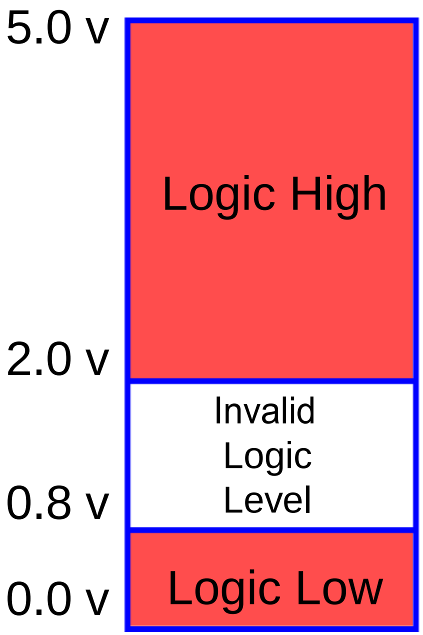
- **Advantages**
  - ✓ Major advantages of the Analog signal is infinite amount of data.
  - ✓ Density is much higher.
  - ✓ Easy processing.
- **Disadvantages**
  - ✓ Unwanted noise in recording.
  - ✓ If we transmit data at long distance then unwanted disturbance is there.
  - ✓ Generation loss is also a big con of analog signals.

Logic

Before examining digital signals, we must define logic levels. A logic level is a voltage level that represents a defined digital state.

✓ Logic HIGH: The higher of two voltages, typically 5 volts

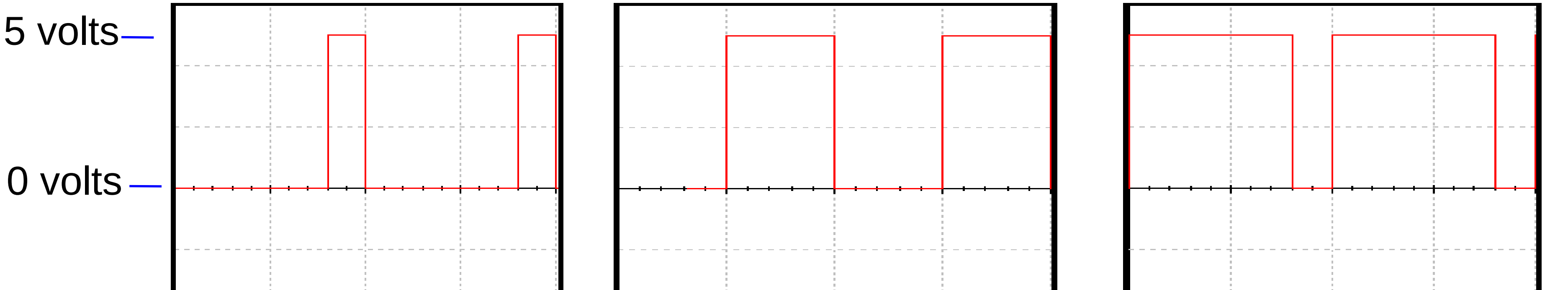
✓ Logic LOW: The lower of two voltages, typically 0 volts



Logic Level	**Voltage**	**True/False**	**On/Off**	**0/1**
HIGH	5 volts	True	On	1
LOW	0 volts	False	Off	0

Examples of Digital

- ✓ Digital signal are commonly referred to as square waves or clock
  signals.
- ✓ Their minimum value must be 0 volts, and their maximum value
  must be 5 volts.
- ✓ They can be periodic (repeating) or non-periodic.
- ✓ The time the signal is high (t_H) can vary anywhere from 1% of the
  period to 99% of the period.
- ✓ Text and Integers.



Parts of Digital

Amplitude:
For digital signals, this will ALWAYS be 5 volts.

Period:
The time it takes for a periodic signal to repeat. (seconds)

Frequency:
A measure of the number of occurrences of the signal per second. (Hertz, Hz)

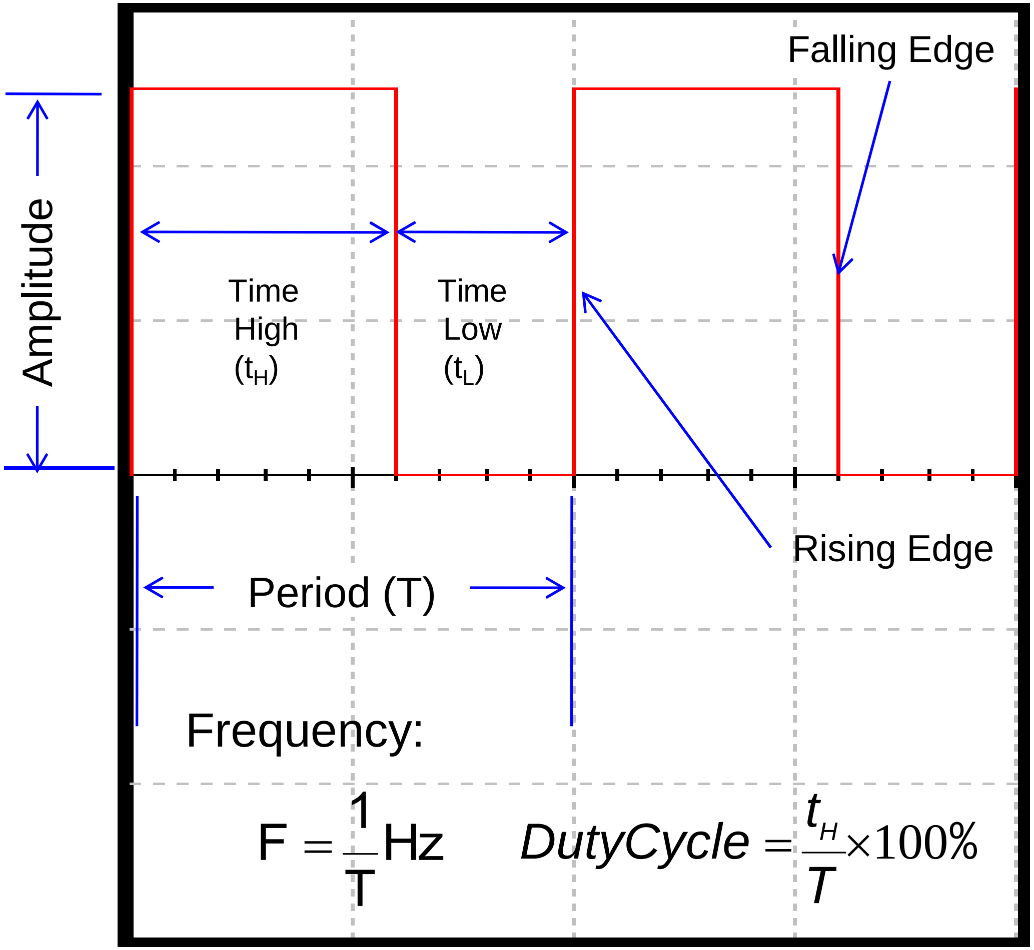
Time High (t_H):
The time the signal is at 5 v.

Time Low (t_L):
The time the signal is at 0 v.

Duty Cycle:
The ratio of t_H to the total period (T).

Rising Edge:
A 0-to-1 transition of the signal.

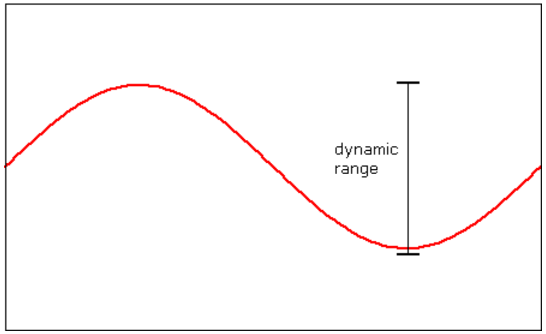
Falling Edge:
A 1-to-0 transition of the signal.



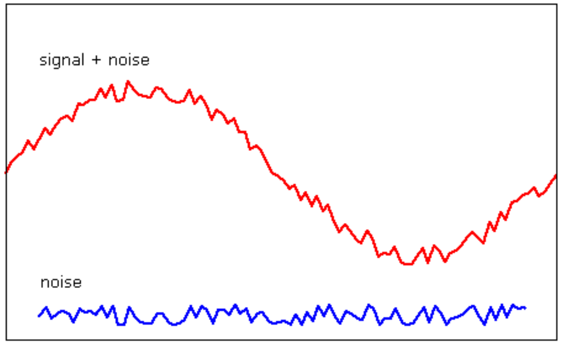
Pros and Cons Digital

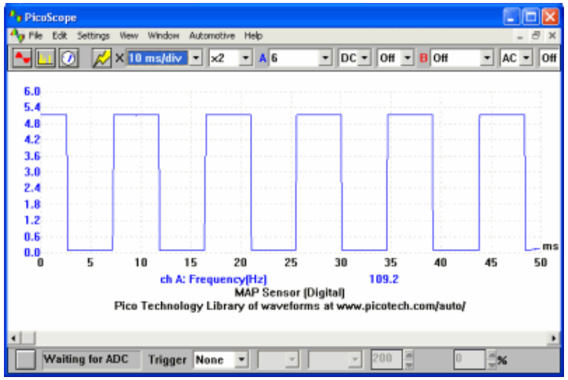
- **Advantages**
  - ✓ Because of their digital nature they can travel faster in over digital lines.
  - ✓ Ability to transfer more data as compared to analog.
- **Disadvantages**
  - ✓ Greater bandwidth is essential.
  - ✓ Systems and processing is more complex.

Example of using digital over

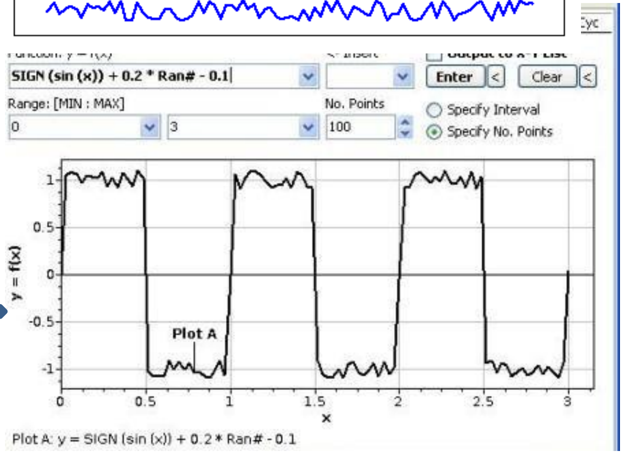


Noisy channel





Noisy channel



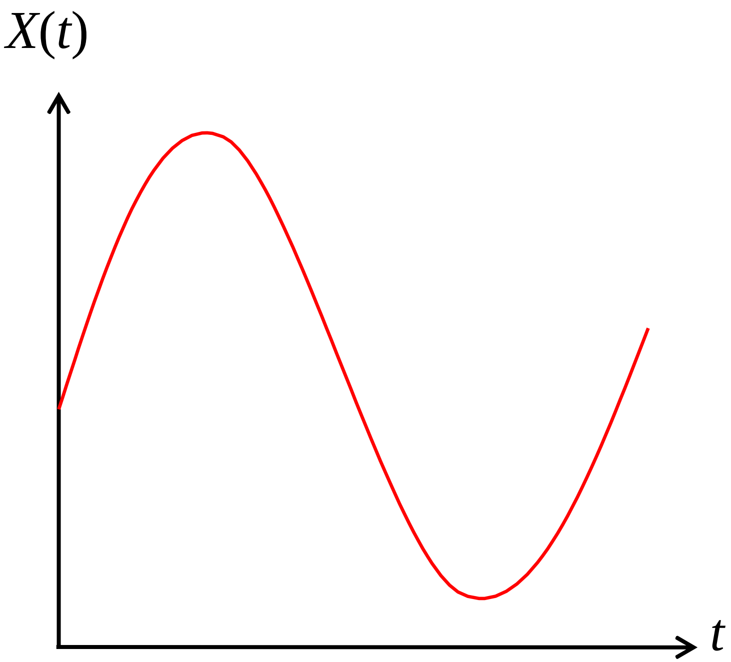
✓ Digital systems are less sensitive to noise

✓ As long as 0 is distinguishable from 1

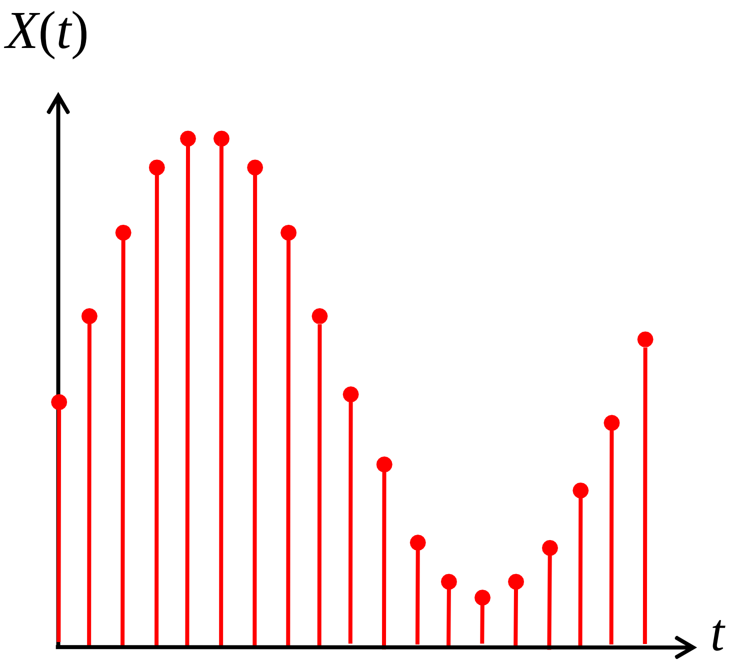
Analog Vs Digital

- Analog system
  - ✓ The physical quantities or signals may vary continuously over a specified range.
- Digital system
  - ✓ The physical quantities or signals can assume only discrete values.
  - ✓ Greater accuracy

Analog Vs Digital



Analog signal



Digital signal

Advantages of Digital System over Analog

- ✓ Digital Systems are easier to design
- ✓ Information storage is easy
- ✓ Accuracy & Precision are greater
- ✓ Digital systems are more versatile
- ✓ Digital circuits are less affected by noise
- ✓ More digital circuitry can be fabricated on IC chips
- ✓ Reliability is more

What is Signal?

- ✓ A signal is a physical quantity (sound, light, voltage, current) that carries **information**
  - – The power cable supplies power but no information (not a signal)
  - – A USB cable carries information (files)
- ✓ Examples of quantities used as digital information signals
  - – Voltage: 5V (logic 1), 0V (logic 0) in digital circuits
  - – Magnetic field orientation in magnetic hard disks
  - – Pits and lands on the CD surface reflect the light from the laser differently, and that difference is encoded as

binary data

Unit I – Number System and Codes

- **Number System: Base or radix of number systems,** Binary, Octal, Decimal and Hexadecimal number system.
- **Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- Subtraction using 1's complement and 2's complement
- **Codes:** BCD, Gray Code, Excess-3, ASCII code
- **BCD Arithmetic:** BCD Addition

Number

- ✓ A number system defines a set of values used to represent quantity.

Different Number

- ✓ Decimal Number System

  - Base/Radix 10

- ✓ Binary Number System

  - Base/Radix 2

- ✓ Octal Number System

  - Base/Radix 8

- ✓ Hexadecimal Number System

  Base/Radix 16

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Binary, Octal, Hexadecimal and **Decimal number system.**
- ✓ **Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓ Subtraction using 1's complement and 2's complement
- ✓ **Codes:** BCD, Gray Code, Excess-3, ASCII code
- ✓ **BCD Arithmetic:** BCD Addition

Decimal Number

- ✓ Decimal number system contains ten unique symbols 0,1,2,3,4,5,6,7,8 and 9
- ✓ Since counting in decimal involves ten symbols, we can say that its base or radix is ten.
- ✓ It is a positional weighted system

Decimal Number

- ✓ In this system, any number (integer, fraction or mixed) of any magnitude can be represented by the use of these ten symbols only

- ✓ Each symbols in the number is called a "**<u>Digit</u>**"

Decimal Number

Structure:

MSD ... LSD

Decimal No. $\ldots\ldots d_3 \quad d_2 \quad d_1 \quad d_0 \;.\; d_{-1} \quad d_{-2} \ldots$

Positional Weights $10^3 \quad 10^2 \quad 10^1 \quad 10^0 \quad 10^{-1} \quad 10^{-2}$

Decimal Point

Decimal Number

- ✓ **MSD:** The leftmost digit in any number representation, which has the greatest positional weight out of all the digits present in that number is called the "**Most Significant Digit**" (MSD)

- ✓ **LSD:** The rightmost digit in any number representation, which has the least positional weight out of all the digits present in that number is called the "**Least Significant Digit**" (LSD)

Decimal Number

- Examples

  1214

  1897

  9875.54

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Hexadecimal, Octal, Decimal and **Binary number system.**
- ✓ **Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓ Subtraction using 1's complement and 2's complement
- ✓ **Codes:** BCD, Gray Code, Excess-3, ASCII code
- ✓ **BCD Arithmetic:** BCD Addition

Binary Number

- ✓ Binary number system is a positional weighted system
- ✓ It contains two unique symbols 0 and 1
- ✓ Since counting in binary involves two symbols, we can say that its base or radix is two.

Binary Number

- ✓A binary digit is called a “**<u>Bit</u>**”
- ✓A binary number consists of a sequence of bits, each of which is either a 0 or a 1.
- ✓The binary point separates the integer and fraction parts

Binary Number

Structure:

		MSB						LSB	
Binary No.	......	b_3	b_2	b_1	b_0	$\cdot$	b_{-1}	b_{-2}	....
Positional Weights		2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	

Binary Point

Binary Number

- ✓ **MSB:** The leftmost bit in a given binary number with the highest positional weight is called the "**Most Significant Bit**" (MSB)

- ✓ **LSB:** The rightmost bit in a given binary number with the lowest positional weight is called the "**Least Significant Bit**" (LSB)

Binary Number

Decimal No.	Binary No.
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

Decimal No.	Binary No.
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

Terms related to Binary

- ✓ **BIT:** The binary digits (0 and 1) are called bits.

  - Single unit in binary digit is called “Bit”

  - Example 1

  0

Terms related to Binary

✓ **NIBBLE:** A nibble is a combination of 4 binary bits.

Examples, 1110

0000

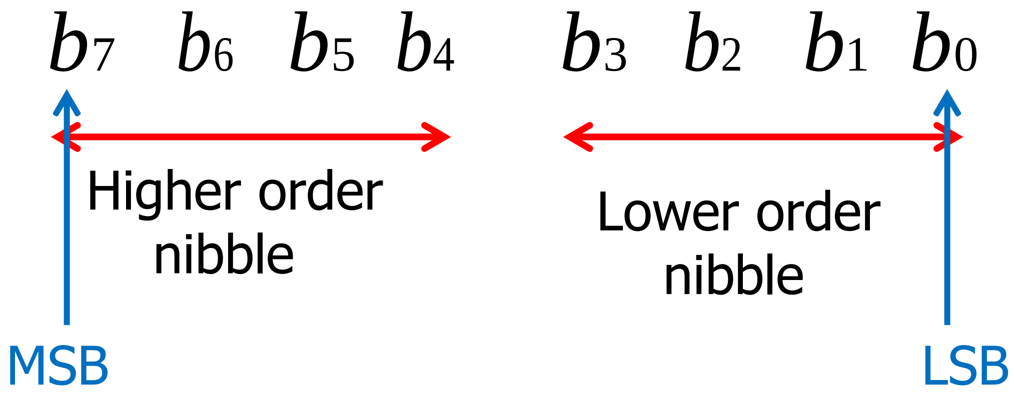
1001

0101

Terms related to Binary

- ✓ **BYTE:** A byte is a combination of 8 binary bits.

  - ✓ The number of distinct values represented by a byte is 256 ranging from 0000 0000 to 1111 1111.



Terms related to Binary

- **WORD:** A word is a combination of 16 binary bits. Hence it consists of two bytes.

$b_{15}\ b_{14}\ b_{13}\ b_{12}\ b_{11}\ b_{10}\ b_{9}\ b_{8}\quad b_{7}\ b_{6}\ b_{5}\ b_{4}\ b_{3}\ b_{2}\ b_{1}\ b_{0}$

Higher order byte

Lower order byte

MSB

LSB

Terms related to Binary

- **DOUBLE WORD:** A double word is exactly what its name implies, two words.

  -It is a combination of 32 binary bits.

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Binary, Hexadecimal, Decimal and **Octal number system.**
- ✓ **Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓ Subtraction using 1's complement and 2's complement
- ✓ **Codes:** BCD, Gray Code, Excess-3, ASCII code
- ✓ **BCD Arithmetic:** BCD Addition

Octal Number

- ✓ Octal number system is a positional weighted system
- ✓ It contains eight unique symbols 0,1,2,3,4,5,6 and 7
- ✓ Since counting in octal involves eight symbols, we can say that its base or radix is eight.

Octal Number

- ✓ The largest value of a digit in the octal system will be 7.

- ✓ That means the octal number higher than 7 will not be 8, instead of that it will be 10.

Octal Number

Structure:

MSD LSD

Octal No. ...... O_3 O_2 O_1 O_0 . O_{-1} O_{-2} ....

Positional Weights 8^3 8^2 8^1 8^0 8^{-1} 8^{-2}

Radix Point

Octal Number

- Since its base $8 = 2^3$, every 3 bit group of binary can be represented by an octal digit.

- An octal number is thus 1/3rd the length of the corresponding binary number

Octal Number

Decimal No.	Binary No.	Octal No.
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	10
9	1001	11
10	1010	12
11	1011	13
12	1100	14

Octal Number

13	1101	15

Unit I – Number System and Codes

- ✓**Number System:** Base or radix of number systems, Binary, Octal, Decimal and **Hexadecimal number system.**
- ✓**Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓Subtraction using 1's complement and 2's complement
- ✓**Codes:** BCD, Gray Code, Excess-3, ASCII code
- ✓**BCD Arithmetic:** BCD Addition

Hexadecimal Number System

- ✓ Binary numbers are long. These numbers are fine for machines but are too lengthy to be handled by human beings. So there is a need to represent the binary numbers concisely.

- ✓ One number system developed with this objective is the hexadecimal number system (or Hex)

Hexadecimal Number System

- ✓ Hex number system is a positional weighted system
- ✓ It contains sixteen unique symbols 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E and F.
- ✓ Since counting in hex involves sixteen symbols, we can say that its base or radix is sixteen.

Hexadecimal Number System

Structure:

		MSD						LSD	
Hex No.	......	H_3	H_2	H_1	H_0	$\cdot$	H_{-1}	H_{-2}	....
Positional Weights		16^3	16^2	16^1	16^0		16^{-1}	16^{-2}	
						Radix Point			

Hexadecimal Number System

- ✓ Since its base $16 = 2^4$, every 4 bit group of binary can

  be represented by an hex digit.

- ✓ An hex number is thus 1/4th the length of the corresponding binary number

- ✓ The hex system is particularly useful for human communications with computer

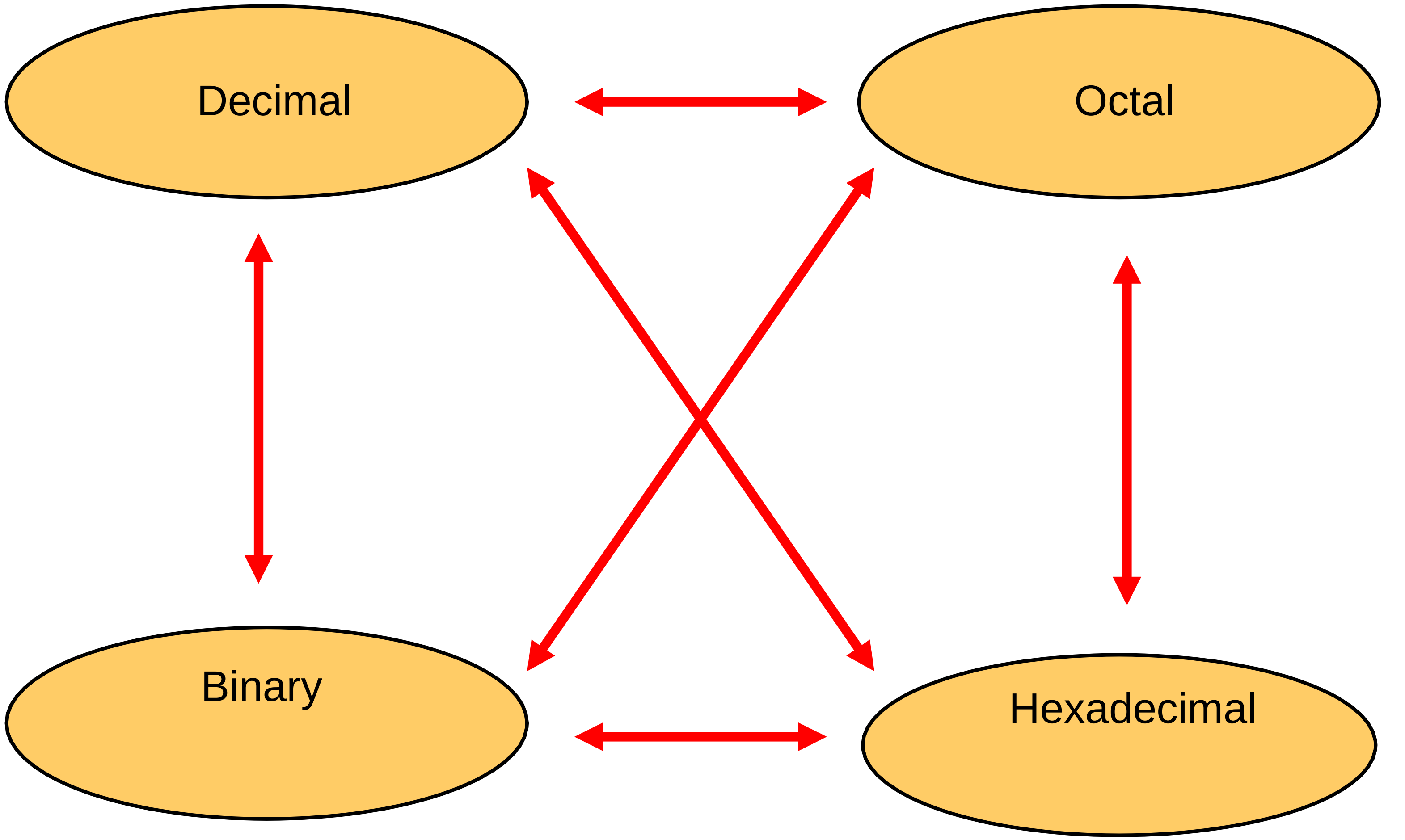
Hexadecimal Number System

Decimal No.	Binary No.	Hex No.
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

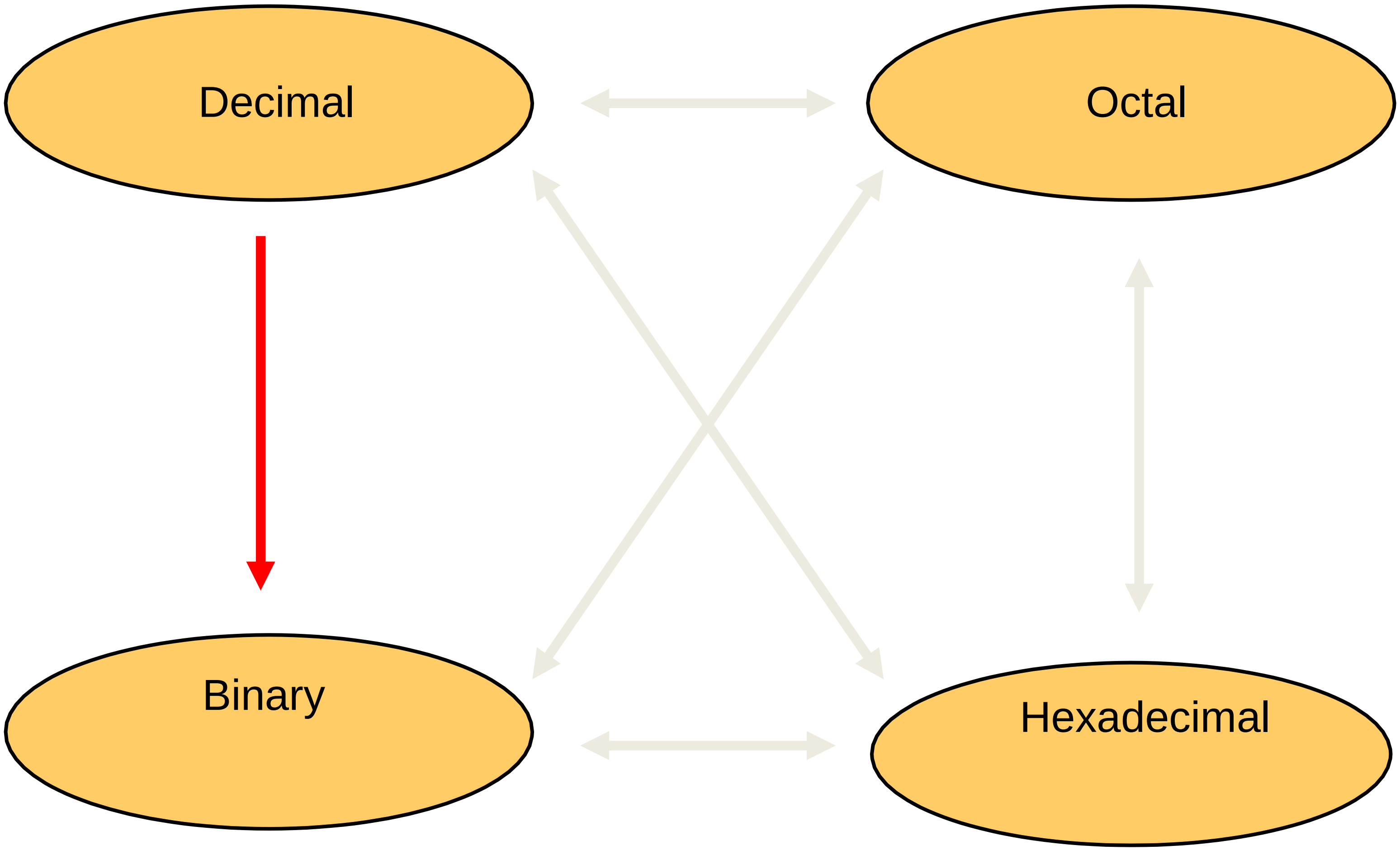
Decimal No.	Binary No.	Hex No.
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Conversion Among

Possibilities



Conversion from Decimal Number to Binary



Conversion of Decimal number into Binary number (Integer Number)

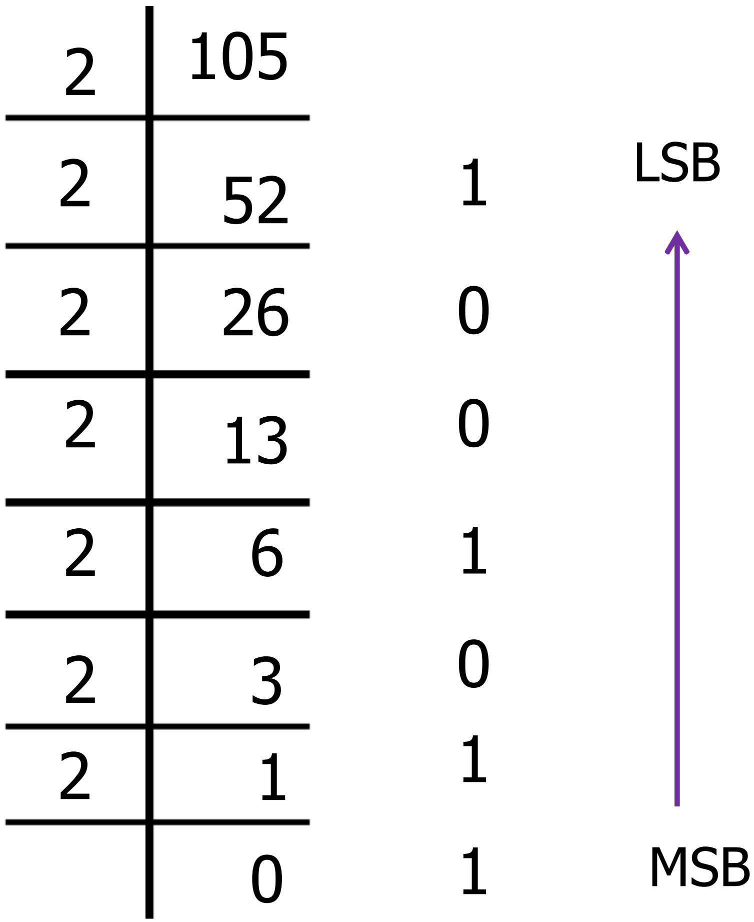
Procedure:

1. Divide the decimal no by the base 2, noting the remainder.
2. Continue to divide the quotient by 2 until there is nothing left, keeping the track of the remainders from each step.

3. List the remainder values in reverse order to find the number’s binary equivalent

Example:-

Convert $(105)_{10}$ to its Binary Number.



$$(105)_{10} = (1101001)_2$$

Conversion of Decimal number into Binary number (Fractional Number)

Procedure:

1. Multiply the given fractional number by base 2.
2. Record the carry generated in this multiplication as MSB.
3. Multiply only the fractional number of the product in step 2 by 2 and record the carry as the next bit to MSB.
4. Repeat the steps 2 and 3 up to 5 bits. The last carry

will represent the LSB of equivalent binary number

Example: Convert 0.42 decimal number in to it's binary number

0.42 X 2	=	0.84	0	MSB
0.84 X 2	=	1.68	1	
0.68 X 2	=	1.36	1	
0.36 X 2	=	0.72	0	
0.72 X 2	=	1.44	1	LSB

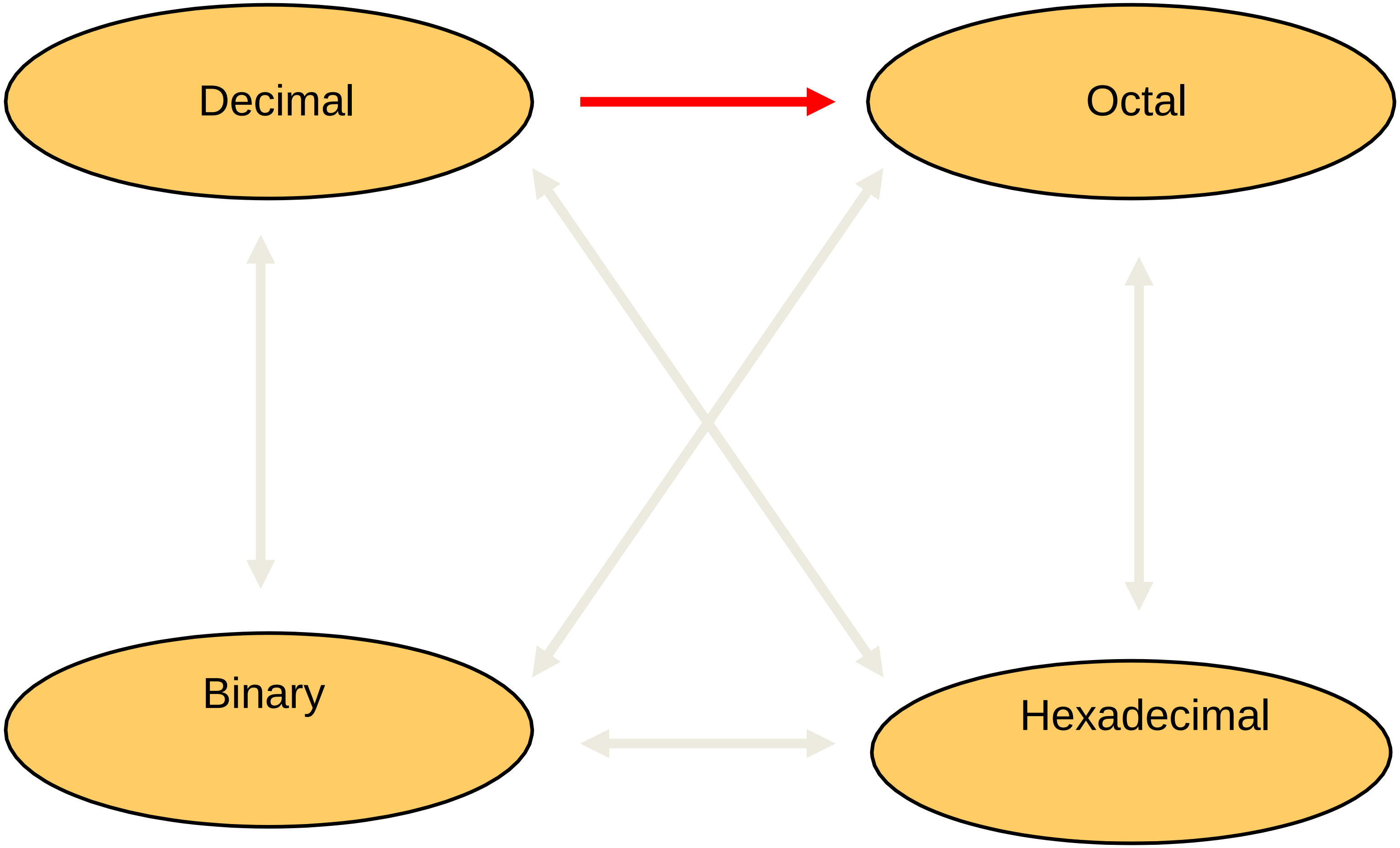
$$(0.42)_{10} = (0.01101)_2$$

Exerci

- Convert following Decimal Numbers in to its equivalent Binary Number:

    1. $(1248.56)_{10} = (?)_2$
    2. $(8957.75)_{10} = (?)_2$
    3. $(420.6)_{10} = (?)_2$
    4. $(8476.47)_{10} = (?)_2$

Conversion from Decimal Number to Octal



Conversion of Decimal Number into Octal Number (Integer Number)

Procedure:

1. Divide the decimal no by the base 8, noting the remainder.

2. Continue to divide the quotient by 8 until there is nothing left, keeping the track of the remainders from each step.

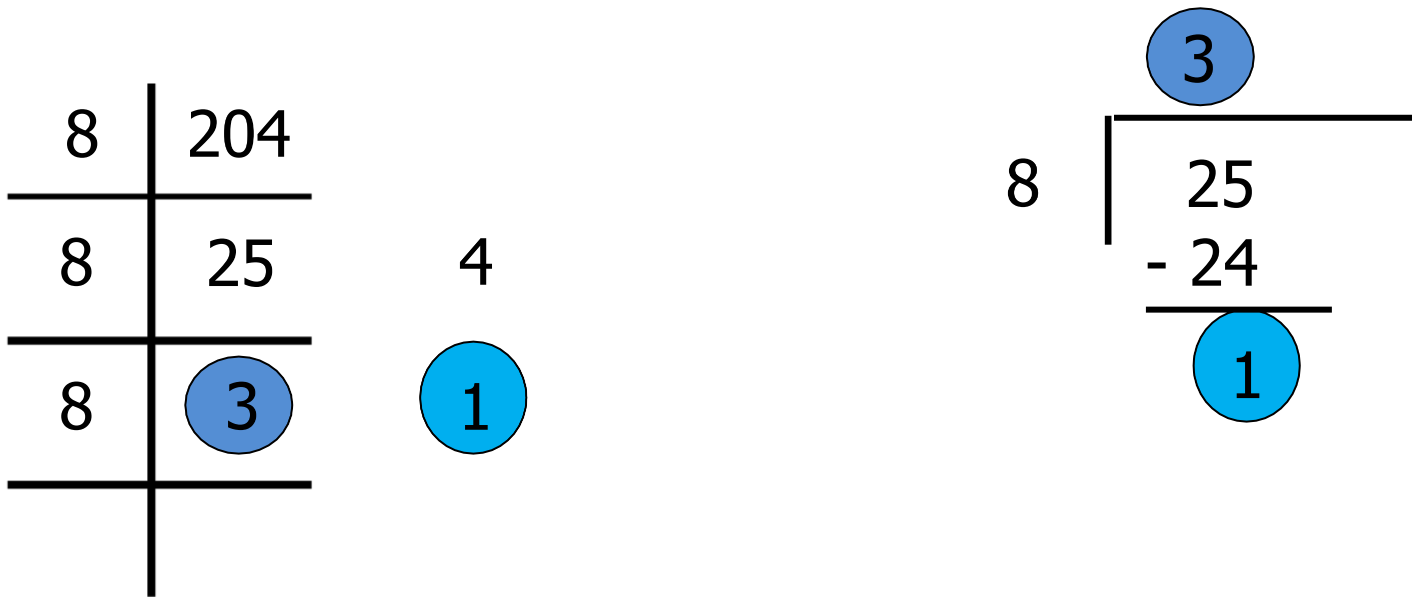
3. List the remainder values in reverse order to find the number's octal equivalent

Example: Convert 204 decimal number in to it’s octal number

8	204
8	25

4

$$
\begin{array}{r|l}
 & \;\;25 \\
\hline
8 & 204 \\
 & -16 \\
\hline
 & \;\;44 \\
 & -\;40 \\
\hline
 & \;\;\;\;4
\end{array}
$$


8
204
8
25
4
8
3
1
3
8
25
- 24
1

8	204	
8	25	4
8	3	1
	0	3

8	204		
8	25	4	LSD
8	3	1	
	0	3	MSD

$$(204)_{10} = (314)_8$$

Conversion of Decimal Number into Octal Number
(Fractional Number)

Procedure:

1. Multiply the given fractional number by base 8.
2. Record the carry generated in this multiplication as MSD.
3. Multiply only the fractional number of the product in step 2 by 8 and record the carry as the next bit to MSD.
4. Repeat the steps 2 and 3 up to 5 bits. The last carry

will represent the LSD of equivalent octal number

Example: Convert 0.6234 decimal number in to it's octal number

0.6234 X 8	=	4.9872	4	MSD
0.9872 X 8	=	7.8976	7	
0.8976 X 8	=	7.1808	7	
0.1808 X 8	=	1.4464	1	
0.4464 X 8	=	3.5712	3	LSD

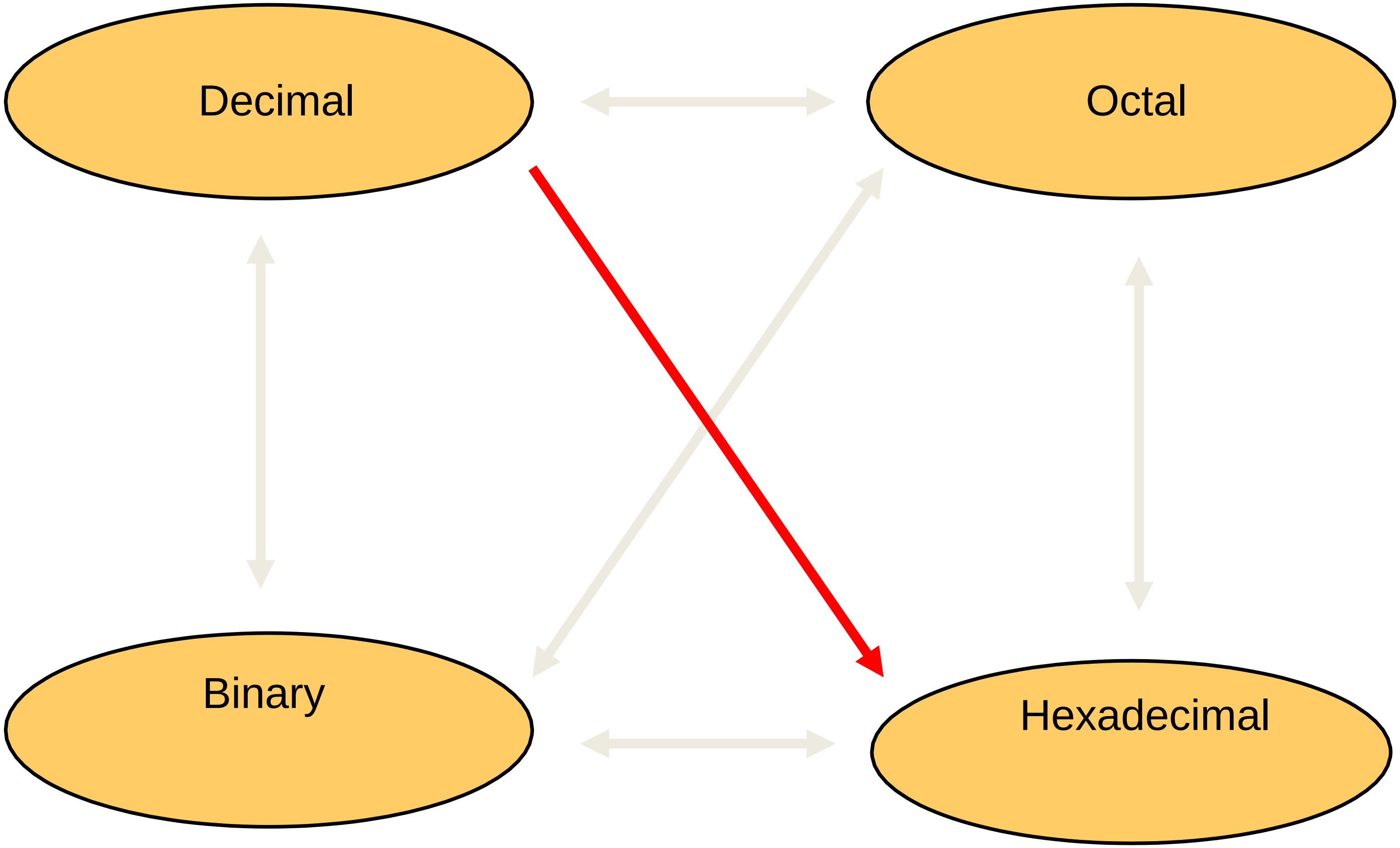
$$(0.6234)_{10} = (0.47713)_8$$

Exerci

- Convert following Decimal Numbers in to its equivalent Octal Number:

    1. $(1248.56)_{10} = (?)_8$
    2. $(8957.75)_{10} = (?)_8$
    3. $(420.6)_{10} = (?)_8$
    4. $(8476.47)_{10} = (?)_8$

Conversion from Decimal Number to Hex



Conversionof Decimal Number into Hexadecimal Number (Integer Number)

Procedure:

1. Divide the decimal no by the base 16, noting the remainder.
2. Continue to divide the quotient by 16 until there is nothing left, keeping the track of the remainders from each step.
3. List the remainder values in reverse order to find the number’s hex equivalent

Example: Convert 2003 decimal number in to it's equivalent Hex number.

16	2003

16	2003
16	125

3

3

$$
\begin{array}{r|l}
 & \quad 1\ \ 2\ \ 5 \\
\hline
16 & \quad 2003 \\
 & -\,16 \\
\cline{2-2}
 & \quad\ \ 40 \\
 & \ -\ 32 \\
\cline{2-2}
 & \quad\ \ \ 83 \\
 & \ \ -\ 80 \\
\cline{2-2}
 & \quad\ \ \ \ 3
\end{array}
$$

16	2003		
16	125	3	3
16	7	13	D

$$\begin{array}{r|l} & 7 \\ \hline 16 & 125 \\ & -\ 112 \\ \cline{2-2} & 13 \end{array}$$

16	2003		
16	125	3	3
16	7	13	D
	0	7	7

16	2003			
16	125	3	3	LSD
16	7	13	D	
	0	7	7	MSD

$$(2003)_{10} = (7D3)_{16}$$

Conversion of Decimal Number into Hexadecimal Number (Fractional Number)

Procedure:

1. Multiply the given fractional number by base 16.
2. Record the carry generated in this multiplication as MSD.
3. Multiply only the fractional number of the product in step 2 by 16 and record the carry as the next bit to MSD.
4. Repeat the steps 2 and 3 up to 5 bits. The last carry will represent the LSD of equivalent hex number

Example: Convert 0.122 decimal number in to it’s hexadecimal number

0.122 X 16 = 1.952	1	1	MSD
0.952 X 16 = 15.232	15	F	
0.232 X 16 = 3.712	3	3	
0.712 X 16 = 11.392	11	B	
0.392 X 16 = 6.272	6	6	LSD

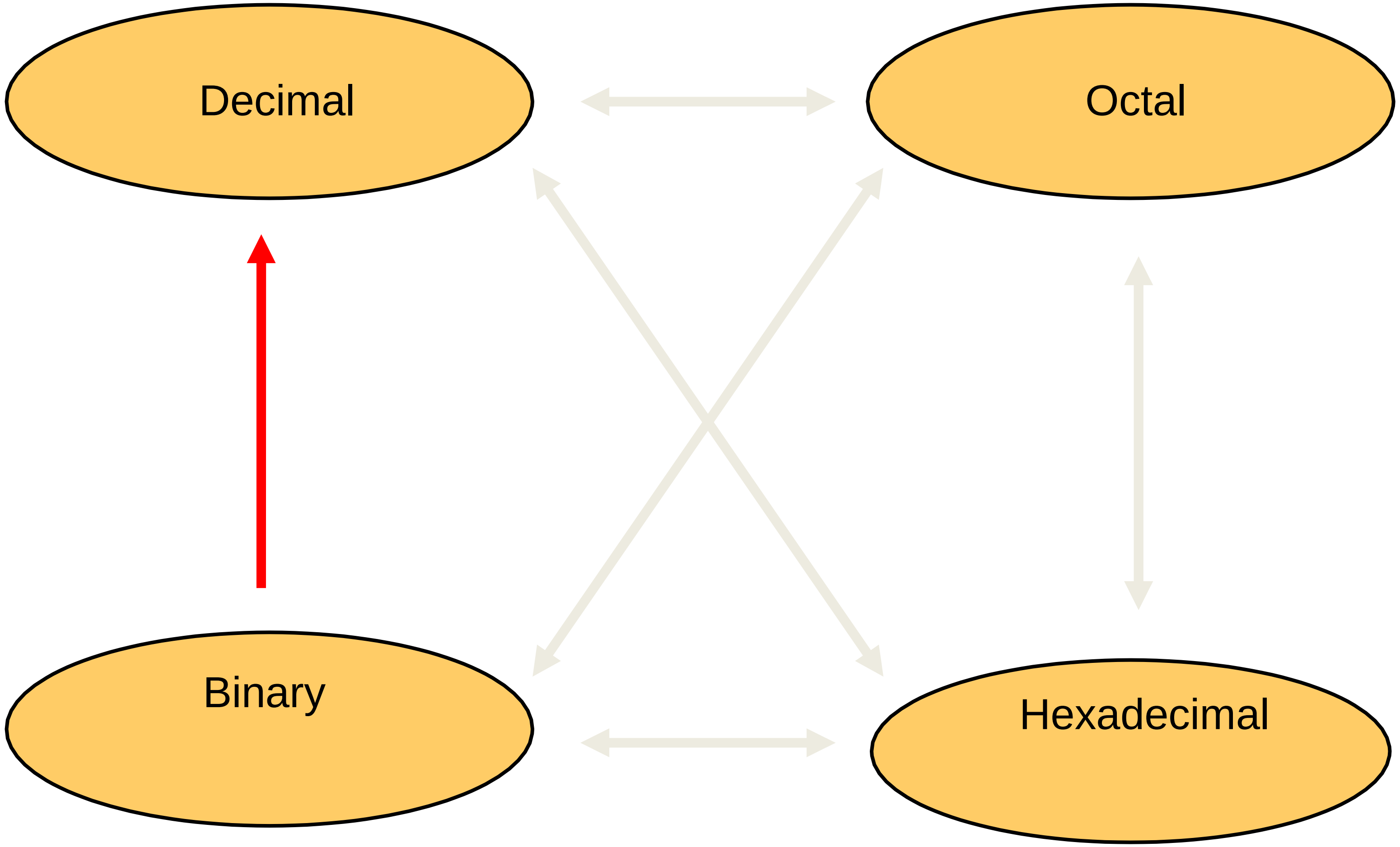
$$(0.122)_{10} = (0.1F3B6)_{16}$$

Exerci

- Convert following Decimal Numbers in to its equivalent Hex Number:

1. $(1248.56)_{10} = (?)_{16}$
2. $(8957.75)_{10} = (?)_{16}$
3. $(420.6)_{10} = (?)_{16}$
4. $(8476.47)_{10} = (?)_{16}$

Conversion from Binary Number to Decimal



Conversion of Binary Number into Decimal

Procedure:

1. Write down the binary number.
2. Write down the weights for different positions.
3. Multiply each bit in the binary number with the corresponding weight to obtain product numbers to get the decimal numbers.
4. Add all the product numbers to get the decimal equivalent

Example: Convert 1011.01 binary number in to it's decimal number

Binary No.	1	0	1	1	·	0	1
Positional Weights	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}

$$=(1\times 2^3)+(0\times 2^2)+(1\times 2^1)+(1\times 2^0).(0\times 2^{-1})+(1\times 2^{-2})$$

$$= 8 + 0 + 2 + 1 . 0 + 0.25$$

$$= 11.25$$

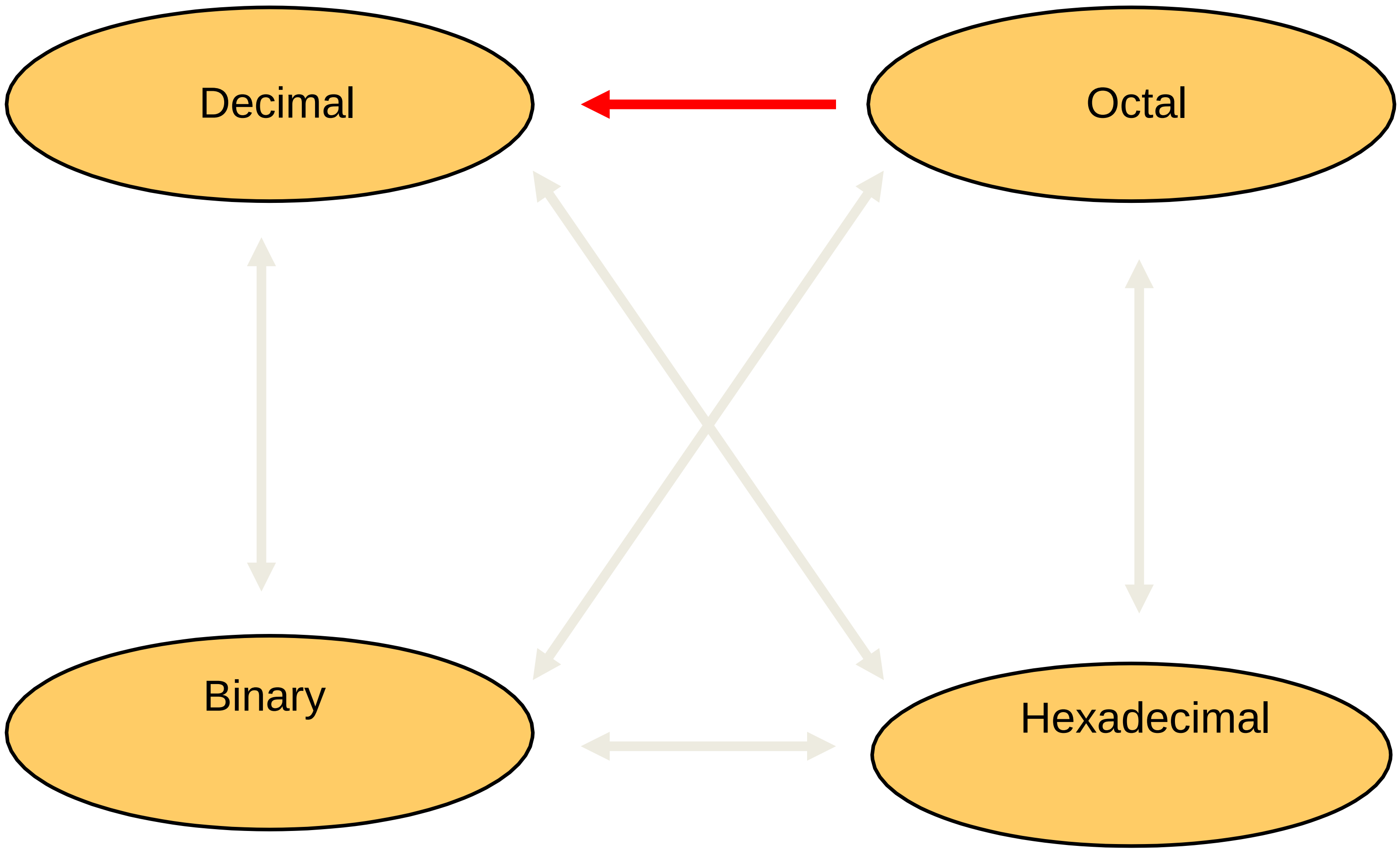
$$\boxed{(1011.01)_2 = (11.25)_{10}}$$

Exerci

- Convert following Binary Numbers in to its equivalent Decimal Number:

1. $(1101110.011)_2 = (?)_{10}$
2. $(1101.11)_2 = (?)_{10}$
3. $(10001.01)_2 = (?)_{10}$

Conversion from Octal Number to Decimal



Conversion of Octal Number into Decimal

Procedure:

1. Write down the octal number.
2. Write down the weights for different positions.
3. Multiply each bit in the binary number with the corresponding weight to obtain product numbers to get the decimal numbers.
4. Add all the product numbers to get the decimal equivalent

Example: Convert 365.24 octal number in to it's decimal number

Octal No.	3	6	5	.	2	4
Positional Weights	8^2	8^1	8^0		8^{-1}	8^{-2}

$$=(3\times8^2)+(6\times8^1)+(5\times8^0).(2\times8^{-1})+(4\times8^{-2})$$

= 192 + 48 + 5 . 0.25 +0.0625

= 245.3125

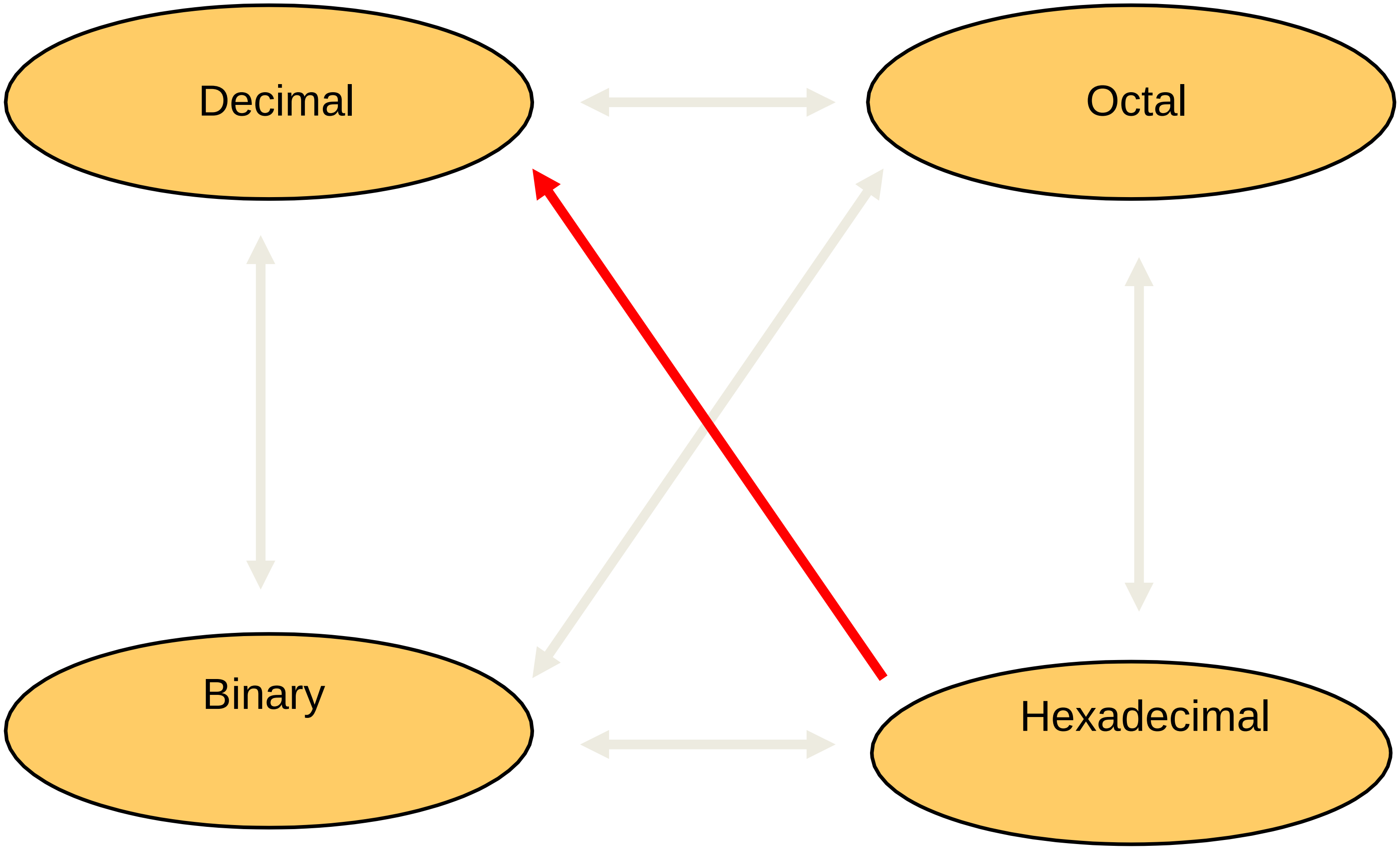
$$\boxed{(365.24)_8=(245.3125)_{10}}$$

Exerci

- Convert following Octal Numbers in to its equivalent Decimal Number:

1. $(3006.05)_8 = (?)_{10}$
2. $(273.56)_8 = (?)_{10}$
3. $(6534.04)_8 = (?)_{10}$

Conversion from Hex Number to Decimal



Conversion of Hexadecimal Number into Decimal

Procedure:

1. Write down the hex number.
2. Write down the weights for different positions.
3. Multiply each bit in the binary number with the corresponding weight to obtain product numbers to get the decimal numbers.
4. Add all the product numbers to get the decimal equivalent

Example: Convert 5826 hex number in to it’s equivalent decimal number.

Hex No.	5	8	2	6
Positional Weights	16^3	16^2	16^1	16^0

$$=(5\times 16^3)+(8\times 16^2)+(2\times 16^1)+(6\times 16^0)$$

=20480 + 2048 + 32 + 6

= 22566

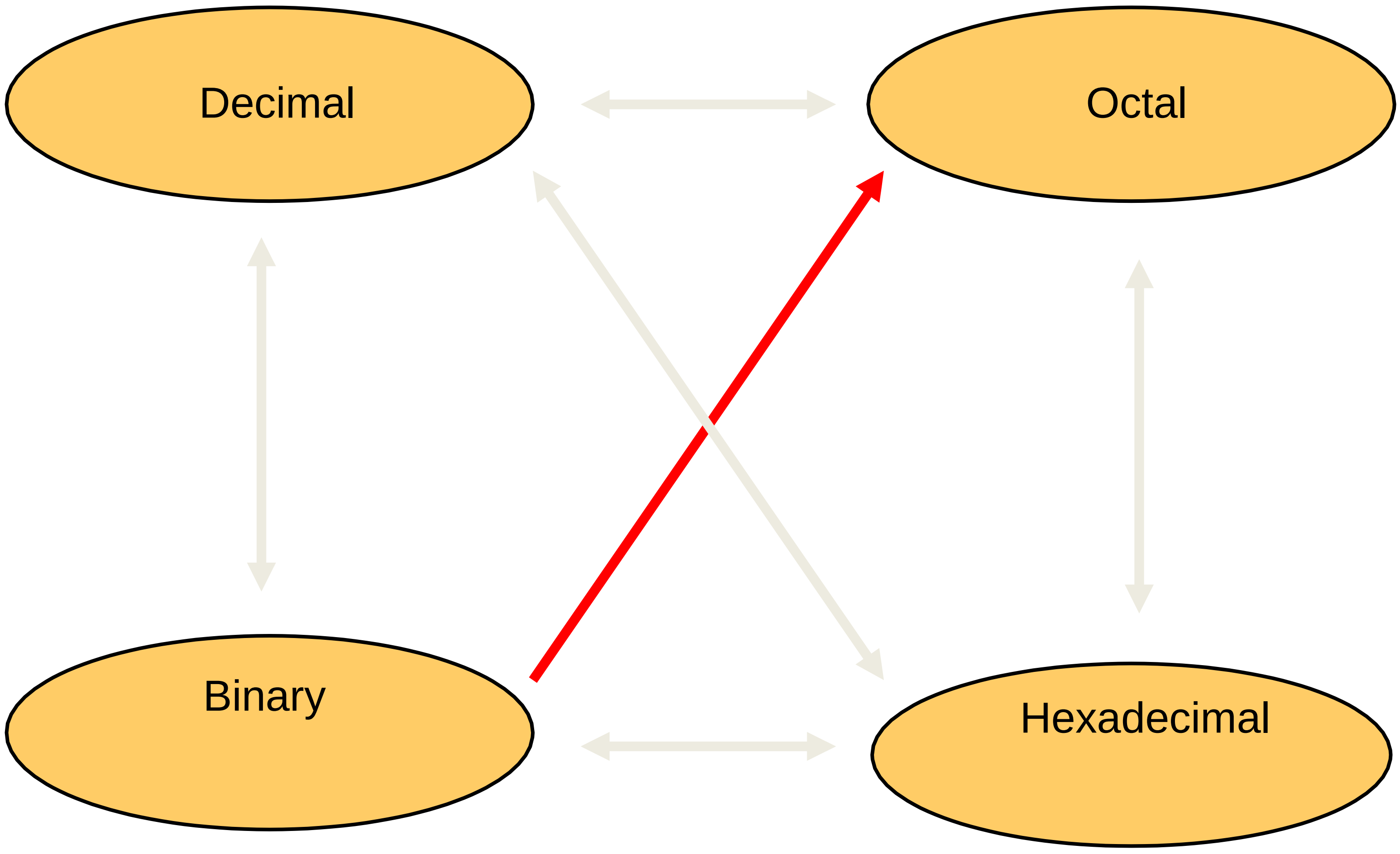
$$(5826)_{16}=(22566)_{10}$$

Exerci

- Convert following Hexadecimal Numbers in to its equivalent Decimal Number:

1. $(4056)_{16} = (\ ?\ )_{10}$
2. $(6B7)_{16} = (\ ?\ )_{10}$
3. $(8E47.AB)_{16} = (\ ?\ )_{10}$

Conversion from Binary Number to Octal



Conversion of Binary Number into Octal

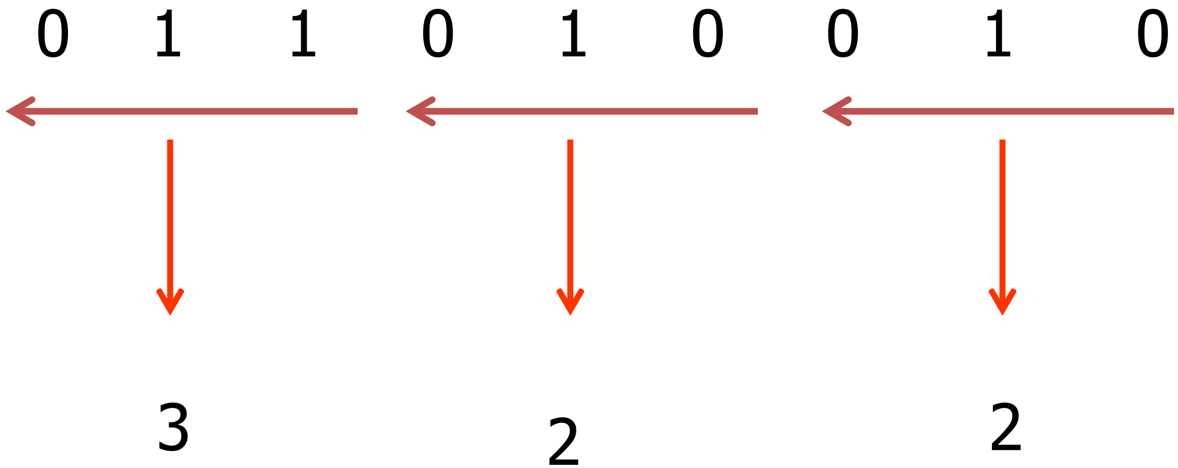
Procedure:

1. Group the binary bits into groups of 3 starting

   from LSB.

2. Convert each group into its equivalent decimal. As the number of bits in each group is restricted to 3, the decimal number will be same as octal number

Example: Convert 11010010 binary number in to it's octal number.



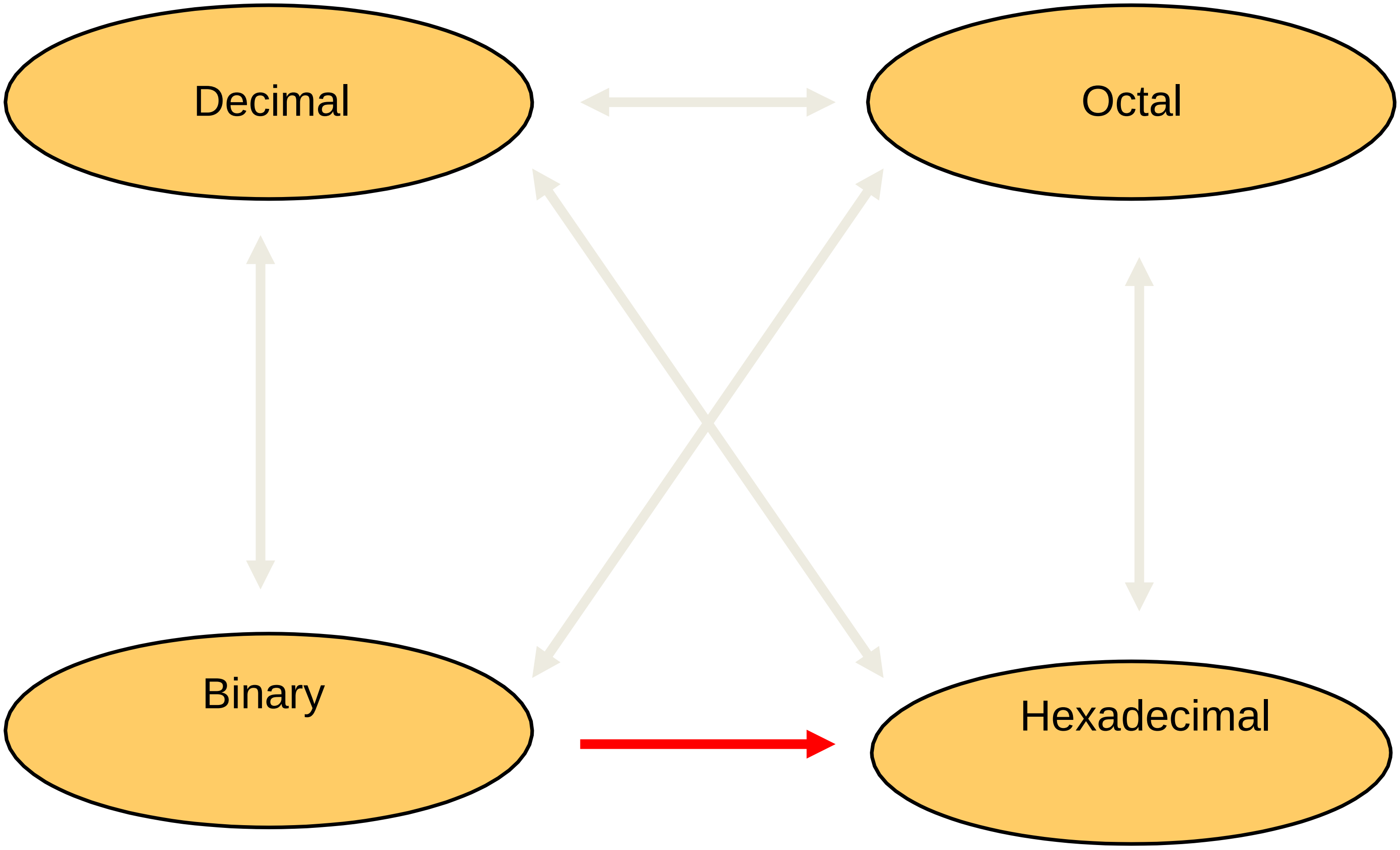
$$(11010010)_2 = (322)_8$$

Exercise

- Convert following Binary Numbers in to its equivalent Octal Number:

1. $(1101110.011)_2 = (?)_8$
2. $(1101.11)_2 = (?)_8$
3. $(10001.01)_2 = (?)_8$

Conversion from Binary Number to Hexadecimal Number



Conversion of Binary Number to Hexadecimal

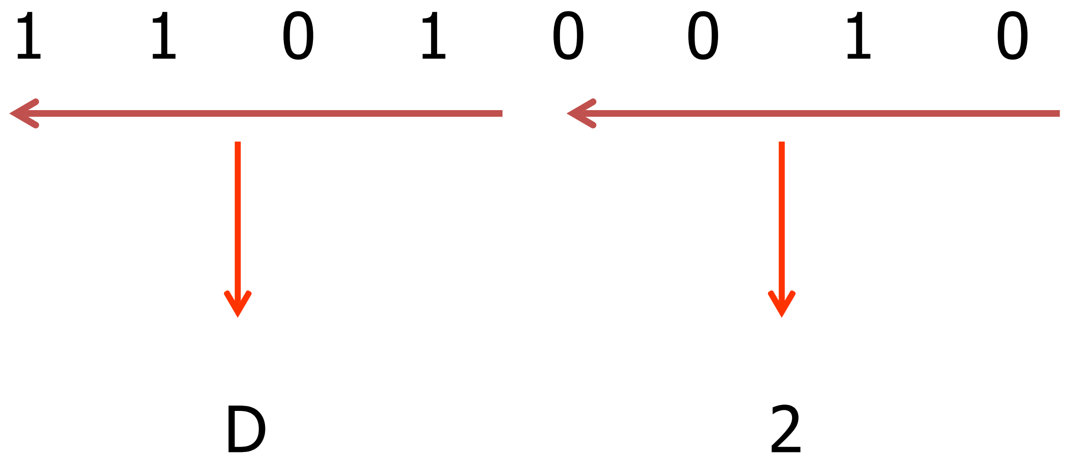
Procedure:

1. Group the binary bits into groups of 4 starting

   from LSB.

2. Convert each group into its equivalent decimal. As the number of bits in each group is restricted to 4, the decimal number will be same as hex number

Example: Convert 11010010 binary number in to it's hexadecimal number.



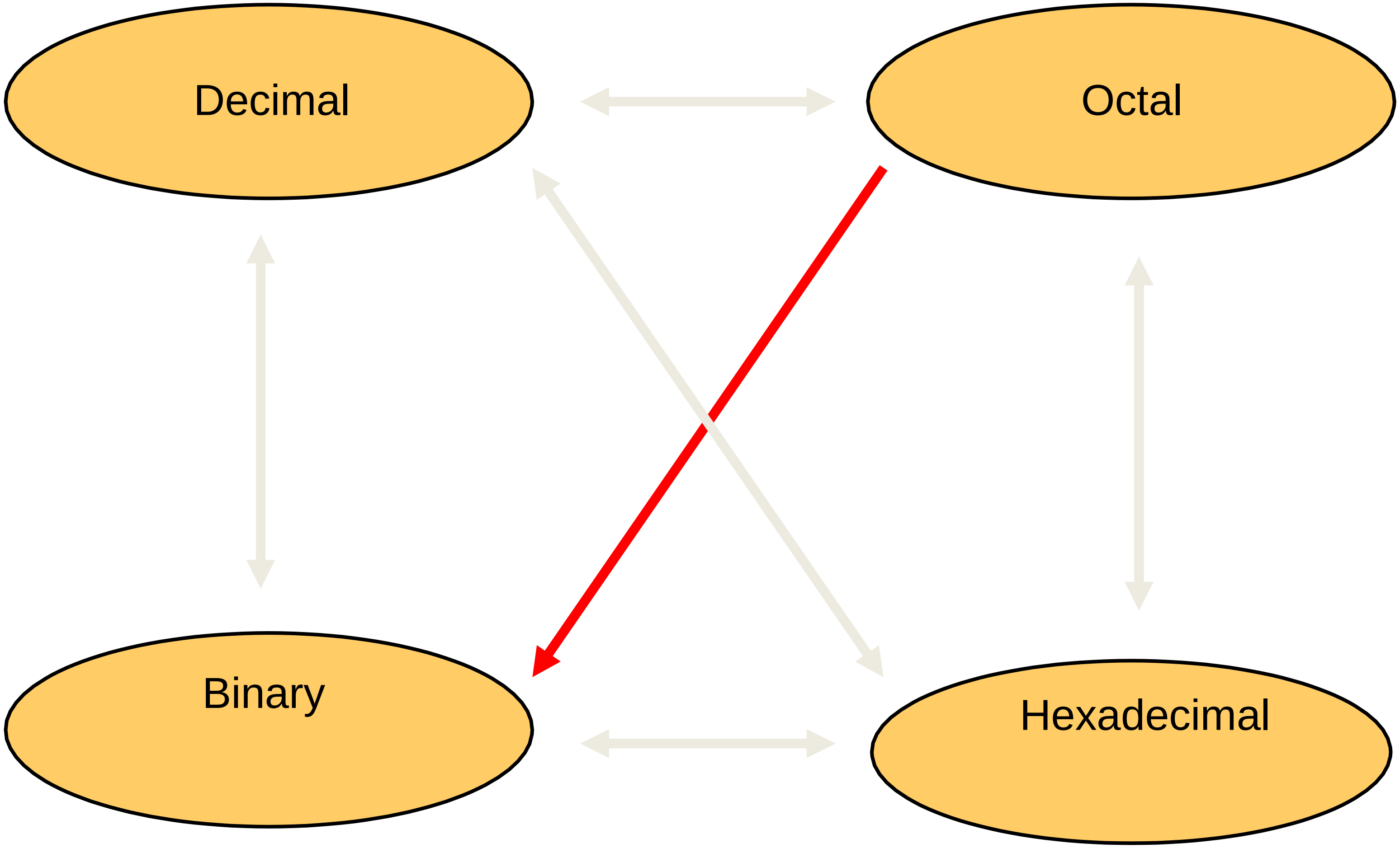
$$\boxed{(11010010)_2 = (D2)_{16}}$$

Exerci

- Convert following Binary Numbers in to its equivalent Hexadecimal Number:

1. $(1101110.011)_2 = (?)_{16}$
2. $(1101.11)_2 = (?)_{16}$
3. $(10001.01)_2 = (?)_{16}$

Conversion from Octal Number to Binary



Conversion of Octal Number into Binary

- ✓ To get the binary equivalent of the given octal number we have to convert each octal digit into its equivalent 3 bit binary number

Example: Convert 364 octal number in to it’s equivalent binary number.

3	6	4
↓	↓	↓
011	110	100

$$(364)_8 = (011110100)_2$$

OR

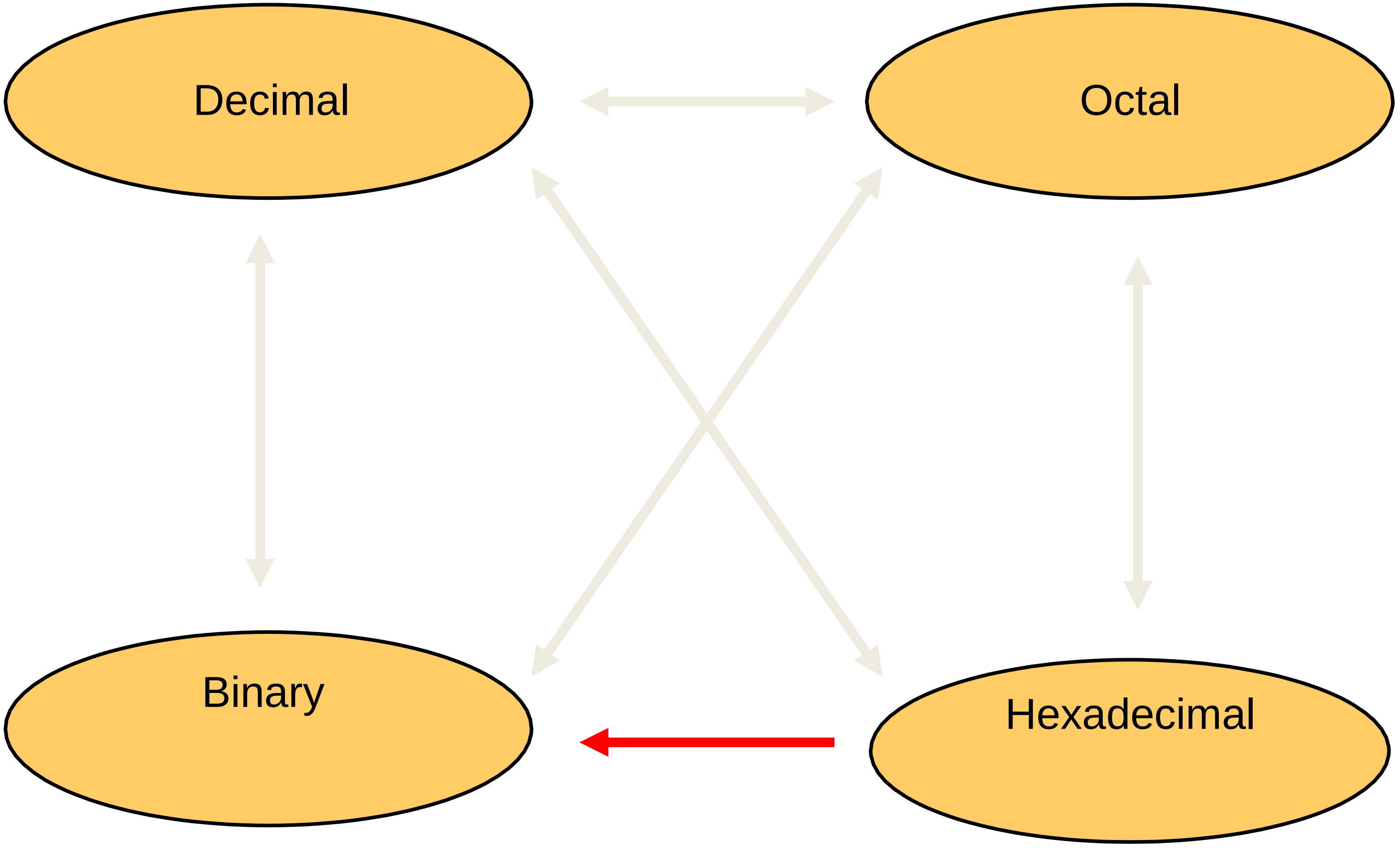
$$\boxed{(364)_8 = (11110100)_2}$$

Exerci

- Convert following Octal Numbers in to its equivalent Binary Number:

1. $(3006.05)_8 = (?)_2$
2. $(273.56)_8 = (?)_2$
3. $(6534.04)_8 = (?)_2$

Conversion from Hex Number to Binary



Conversion of Hexadecimal Number into Binary

- ✓ To get the binary equivalent of the given hex number we have to convert each hex digit into its equivalent 4 bit binary number

Example: Convert AFB2 hex number in to it's equivalent binary number.

A	F	B	2
1010	1111	1011	0010

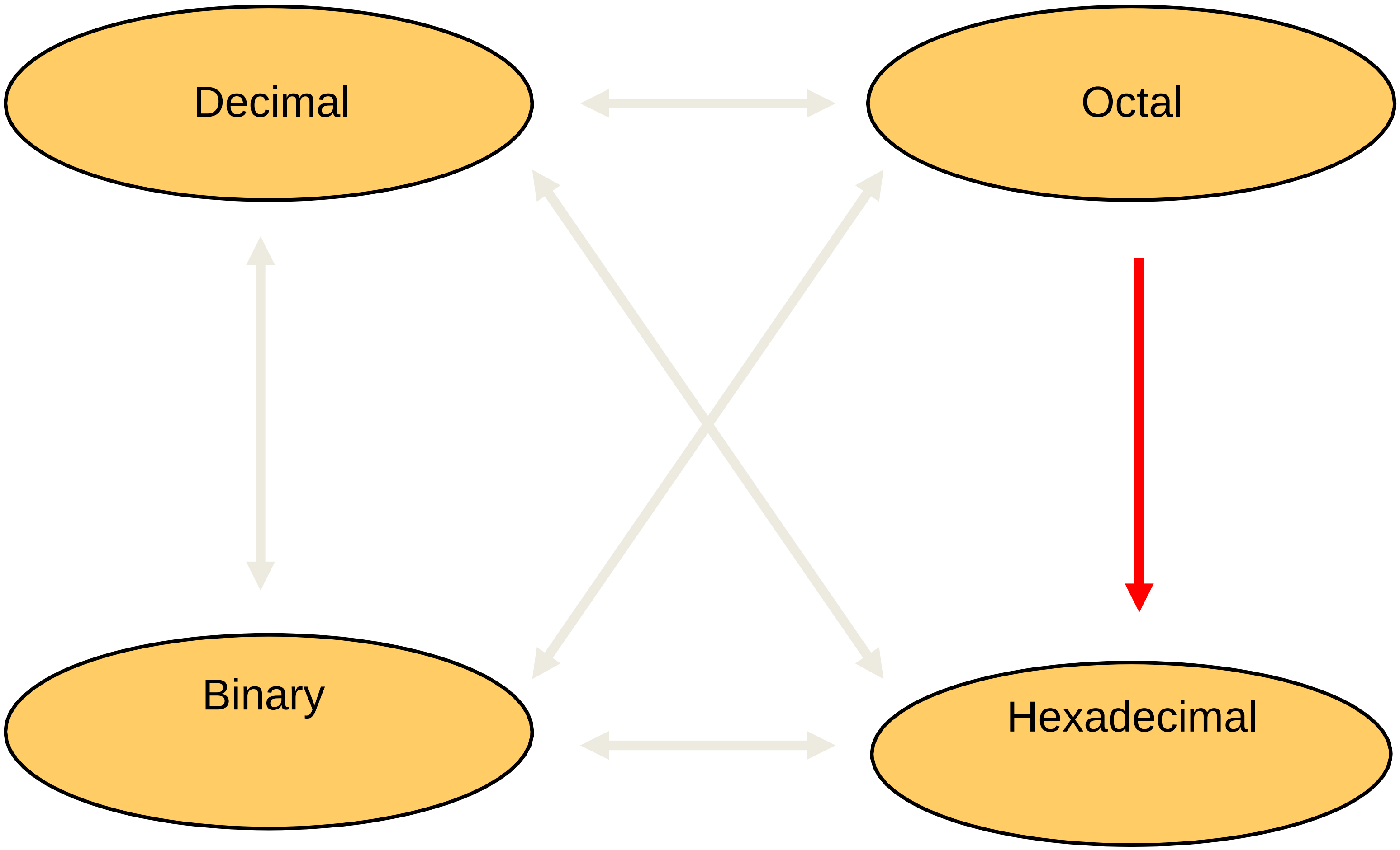
$$(AFB2)_{16} = (1010111110110010)_2$$

Exerci

- Convert following Hexadecimal Numbers in to its equivalent Binary Number:

1. $(4056)_{16} = (?)_2$
2. $(6B7)_{16} = (?)_2$
3. $(8E47.AB)_{16} = (?)_2$

Conversion from Octal Number to Hex



Conversion of Octal Number into Hexadecimal

- ✓ To get hex equivalent number of given octal number, first we have to convert octal number into its 3 bit binary equivalent and then convert binary number into its hex equivalent.

Example: Convert 364 octal number in to it’s equivalent hex number.

3	6	4	Octal Number
011	110	100	Binary Number

011110100 Binary Number

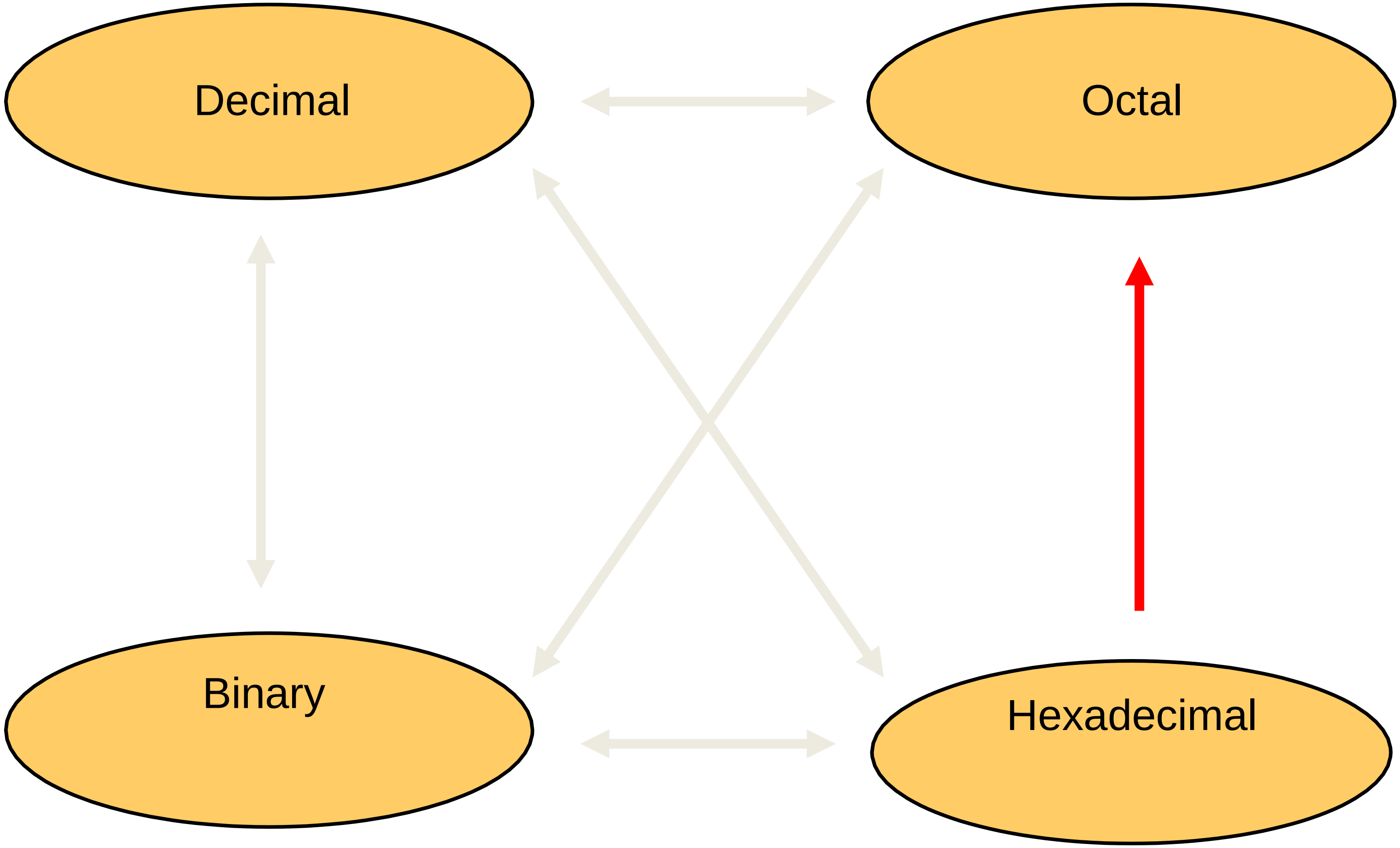
0 F 4 Hex Number

$$(364)_8 = (F4)_{16}$$

Exerci

- Convert following Octal Numbers in to its equivalent Hex Number:
    1. $(3006.05)_8 = (?)_{16}$
    2. $(273.56)_8 = (?)_{16}$
    3. $(6534.04)_8 = (?)_{16}$

Conversion from Hex Number to Octal



Conversion of Hexadecimal Number into Octal

- ✓ To get octal equivalent number of given hex number, first we have to convert hex number into its 4 bit binary equivalent and then convert binary number into its octal equivalent.

Example: Convert 4CA hexadecimal number in to it's octal number.

4	C	A	Hex Number
0100	1100	1010	Binary Number

010011001010 Binary Number

2 3 1 2 Octal Number

4 C A Hex Number

0100 1100 1010 Binary Number

010011001010 Binary Number

2 3 1 2 Octal Number

$$(4CA)_{16} = (2312)_8$$

Exercise

- Convert following Hexadecimal Numbers in to its equivalent Octal Number:

  1. $(4056)_{16} = (\ ?\ )_8$
  2. $(6B7)_{16} = (\ ?\ )_8$
  3. $(8E47.AB)_{16} = (\ ?\ )_8$

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- ✓ **Binary arithmetic: Addition,** Subtraction, Multiplication, Division.
- ✓ Subtraction using 1's complement and 2's complement
- ✓ **Codes:** BCD, Gray Code, Excess-3, ASCII code
- ✓ **BCD Arithmetic:** BCD Addition

Binary

- Following are the four most basic cases for binary addition

0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	10 i.e. 0 with carry 1

Binary

Example: Perform $(10111)_2 + (11001)_2$

Binary

Example: Perform $(10111)_2 + (11001)_2$

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

Binary

Example: Perform $(10111)_2 + (11001)_2$

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

Binary

Example: Perform $(10111)_2 + (11001)_2$

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

Binary

Example: Perform $(10111)_2 + (11001)_2$

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

Binary

Example: Perform $(10111)_2 + (11001)_2$

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

Binary

Example: Perform $(10111)_2 + (11001)_2$

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

$$(10111)_2 + (11001)_2 = (110000)_2$$

Binary

Example: Perform $(1101.101)_2 + (111.011)_2$

Binary

Example: Perform $(1101.101)_2+(111.011)_2$

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

$$(1101.101)_2+(111.011)_2=(10101.000)_2$$

Exerci

- Perform Binary Addition of following:

1. $(11011)_2+(1101)_2$
2. $(1011)_2+(1101)_2+(1001)_2+(1111)_2$
3. $(1010.11)_2+(1101.10)_2+(1001.11)_2+(1111.11)_2$
4. $(10111.101)_2+(110111.01)_2$

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- ✓ **Binary arithmetic:** Addition, **Subtraction,** Multiplication, Division.
- ✓ Subtraction using 1's complement and 2's complement
- ✓ **Codes:** BCD, Gray Code, Excess-3, ASCII code
- ✓ **BCD Arithmetic:** BCD Addition

Binary

- Following are the four most basic cases for binary subtraction

	Subtraction	Borrow
0 - 0 = 0		0
0 - 1 = 1		1
1 - 0 = 1		0
1 - 1 = 0		0

Binary

Example: Perform $(1010.010)_2 - (111.111)_2$

Binary

Example: Perform $(1010.010)_2 - (111.111)_2$

		1		1		1	10	
	0	10	0	10		10	0	10
	1	0	1	0	.	0	1	0
−		1	1	1	.	1	1	1
	0	0	1	0	.	0	1	1

$$(1010.010)_2 + (111.111)_2 = (0010.011)_2$$

Exerci

- Perform Binary Subtraction of following:

1. $(1011)_2 - (101)_2$

2. $(1100.10)_2 - (111.01)_2$

3. $(10110)_2 - (1011)_2$

4. $(10001.01)_2 - (1111.11)_2$

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- ✓ **Binary arithmetic:** Addition, Subtraction, **Multiplication,** Division.
- ✓ Subtraction using 1's complement and 2's complement
- ✓ **Codes:** BCD, Gray Code, Excess-3, ASCII code
- ✓ **BCD Arithmetic:** BCD Addition

Binary

- Followingare the four most basiccases for binary multiplication

0	X	0	=	0
0	X	1	=	0
1	X	0	=	0
1	X	1	=	1

Binary

Example: Perform $(1001)_2 + (1000)_2$

Binary

Example: Perform $(1001)_2 + (1000)_2$

				1	0	0	1
×				1	0	0	0
				0	0	0	0
+			0	0	0	0	x
		0	0	0	0	x	x
	1	0	0	1	x	x	x
	1	0	0	1	0	0	0

$$(1001)_2 + (1000)_2 = (1001000)_2$$

Exerci

- Perform Binary Multiplication of following:

1. $(1101)_2 \times (101)_2$
2. $(1101.11)_2 \times (101.1)_2$
3. $(11001)_2 \times (10)_2$
4. $(10110)_2 \times (10.1)_2$

Unit I – Number System and Codes

- **Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- **Binary arithmetic:** Addition, Subtraction, Multiplication, **Division.**
- Subtraction using 1's complement and 2's complement
- **Codes:** BCD, Gray Code, Excess-3, ASCII code
- **BCD Arithmetic:** BCD Addition

Binary

Example: Perform $(110110)_2 / (101)_2$

Binary

Example: Perform $(110110)_2 \,/\, (101)_2$

```
          1  0  1  0
       ________________
101   |  1  1  0  1  1  0
        -1  0  1
       ____________
         0  0  1  1
              -0  0
            ____________
               1  1  1
              -1  0  1
              ____________
               0  1  0  0
                 -0  0  0
                 ____________
                  1  0  0
```

Exerci

- Perform Binary Division of following:

1. $(1010)_2$ by $(11)_2$
2. $(11110)_2$ by $(101)_2$
3. $(11011)_2$ by $(10.1)_2$
4. $(110111.1)_2$ by $(101)_2$

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- ✓ **Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓ **Subtraction using 1's complement and 2's complement**
- ✓ **Codes:** BCD, Gray Code, Excess-3, ASCII code
- ✓ **BCD Arithmetic:** BCD Addition

1's

- The 1's complement of a number is obtained by simply complementing each bit of the number that is by changing all 0's to 1's and all 1's to 0's.
- This system is called as 1's complement because the number can be subtracted from 1 to obtain

1’s

result

1's

Example: Obtain 1's complement of the 1010

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

1's complement of the 1010 is 0101

1's

Sr. No.	Binary Number	1's Complement
1	1101 0101	0010 1010
2	1001	0110
3	1011 1111	0100 0000
4	1101 1010 0001	0010 0101 1110
5	1110 0111 0101	0001 1000 1010
6	1011 0100 1001	0100 1011 0110
7	1100 0011 0010	0011 1100 1101
8	0001 0010 1000	1110 1101 0111

1's

Subtraction Using 1's

- In 1's complement subtraction, add the 1's complement of subtrahend to the minuend.
- If there is carry out, bring the carry around and add it to LSB.
- Look at the sing bit (MSB), if this is 0, the result is positive and is in its true binary form.
- If the MSB is 1(whether there is a carry or no carry at all), the result is negative & is in its 1's complement form. So take 1's complement to obtain result.

Subtraction using 1’s

Example: Perform using 1’ complement $(9)_{10} - (4)_{10}$

Subtraction using 1's

Example: Perform using 1′ complement $(9)_{10} - (4)_{10}$

Step 1: Take 1′ complement of $(4)_{10} = (0100)_2$

$$= 1011$$

Step 2: Add 9 with 1′ complement of 4

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

final carry

Result

Step 3:If carry is generated add final carry to the result

Example Continue

		1	0	0	1	
+		1	0	1	1	
final carry	1	0	1	0	0	Result
					1	
		0	1	0	1	Final Result

When the final carry is produced the answer is positive and is in its true binary form

Exercise

- Perform Binary Subtraction using 1's Complement method

  1. $(52)_{10} - (17)_{10}$
  2. $(46)_{10} - (84)_{10}$
  3. $(63.75)_{10}-(17.5)_{10}$
  4. $(73.5)_{10}-(112.75)_{10}$

2’s

- ✓ The 2’s complement of a number is obtained by adding 1 to the 1’s complement of that number

2's

Example: Obtain 2's complement of the 1010

2's

Example: Obtain 2's complement of the 1010

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

---1's complement

----2's complement

2's complement of the 1010 is 0110

2's

Sr. No.	Binary Number	1's Complement	2's Complement
1	1101 0101	0010 1010	0010 1011
2	1001	0110	0111
3	1011 1111	0100 0000	0100 0001
4	1101 1010 0001	0010 0101 1110	0010 0101 1111
5	1110 0111 0101	0001 1000 1010	0001 1000 1011

Subtraction Using 2's Complement

- ✓ In 2's complement subtraction, add the 2's complement of subtrahend to the minuend.
- ✓ If carry is generated then the result is positive and in its true form.
- ✓ If the carry is not produced, then the result is negative and in its 2's complement form.

*Carry is always to be discarded

Subtraction Using 2's

Example: Perform using 2' complement $(9)_{10} - (4)_{10}$

Subtraction Using 2’s

Example: Perform using 2′ complement $(9)_{10} - (4)_{10}$

Step 1: Take 2′ complement of $(4)_{10} = (0100)_2$

$$= 1011 + 1 = 1100$$

Step 2: Add 9 with 2′ complement of 4

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

final carry
Discard

Final Result

If Carry is generated, discard carry. The result is positive and its true binary form

Exercise

- Perform Binary Subtraction using 2's Complement method

1. $(46)_{10} - (19)_{10}$
2. $(27)_{10} - (75)_{10}$
3. $(125.3)_{10}-(46.7)_{10}$
4. $(36.75)_{10}-(89.5)_{10}$

Unit I – Number System and Codes

- ✓**Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- ✓**Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓Subtraction using 1's complement and 2's complement
- ✓**Codes: BCD,** Gray Code, Excess-3, ASCII code
- ✓**BCD Arithmetic:** BCD Addition

BCD or 8421

- ✓ The smallest BCD number is (0000) and the largest is (1001). The next number to 9 will be 10 which is expressed as (0001 0000) in BCD.
- ✓ There are six illegal combinations 1010, 1011, 1100, 1101, 1110 and 1111 in this code i.e. they are not part of the 8421 BCD code

Decimal to BCD

Sr. No.	Decimal Number	BCD Code
1	8	1000
2	47	0100 0111
3	345	0011 0100 0101
4	99	1001 1001
5	10	0001 0000

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- ✓ **Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓ Subtraction using 1's complement and 2's complement
- ✓ **Codes:** BCD, **Gray Code,** Excess-3, ASCII code
- ✓ **BCD Arithmetic:** BCD Addition

Gray

- ✓ The gray code is non-weighted code.
- ✓ It is not suitable for arithmetic operations.
- ✓ It is a cyclic code because successive code words in this code differ in one bit position only i.e. unit distance code

Binary to Gray Code

✓ If an n bit binary number is represented by $B_n, B_{n-1}, \ldots\ldots B_1$ and its gray code equivalent by $G_n, G_{n-1}, \ldots\ldots. G_1$ wher e B_n and G_n are the MSBs, then gray code bits are obtained from the binary code as follows;

$G_n = B_n$	$G_{n-1} = B_n \oplus B_{n-1}$	$G_{n-2} = B_{n-1} \oplus B_{n-2}$	$\cdots\cdots\cdots\cdots$	$G_1 = B_2 \oplus B_1$

Binary to Gray Code

*where the symbol represents Exclusive-OR operation

Binary to Gray Code

Example 1: Convert 1011 Binary Number into Gray Code

Binary to Gray Code

Example 1: Convert 1011 Binary Number into Gray Code

Binary Number	1	0	1	1

Example 1: Continue

Binary Number	1	0	1	1
Gray Code	1			

Example 1: Continue

Binary Number	1	$\oplus$	0	1	1
Gray Code	1		1		

Example 1: Continue

Binary Number 1 0 $\oplus$ 1 1

Gray Code 1 1 1

Example 1: Continue

Binary Number	1	0	1 $\oplus$	1
Gray Code	1	1	1	0

Example 1: Continue

Binary Number 1 $\rightarrow \oplus \leftarrow$ 0 $\rightarrow \oplus \leftarrow$ 1 $\rightarrow \oplus \leftarrow$ 1

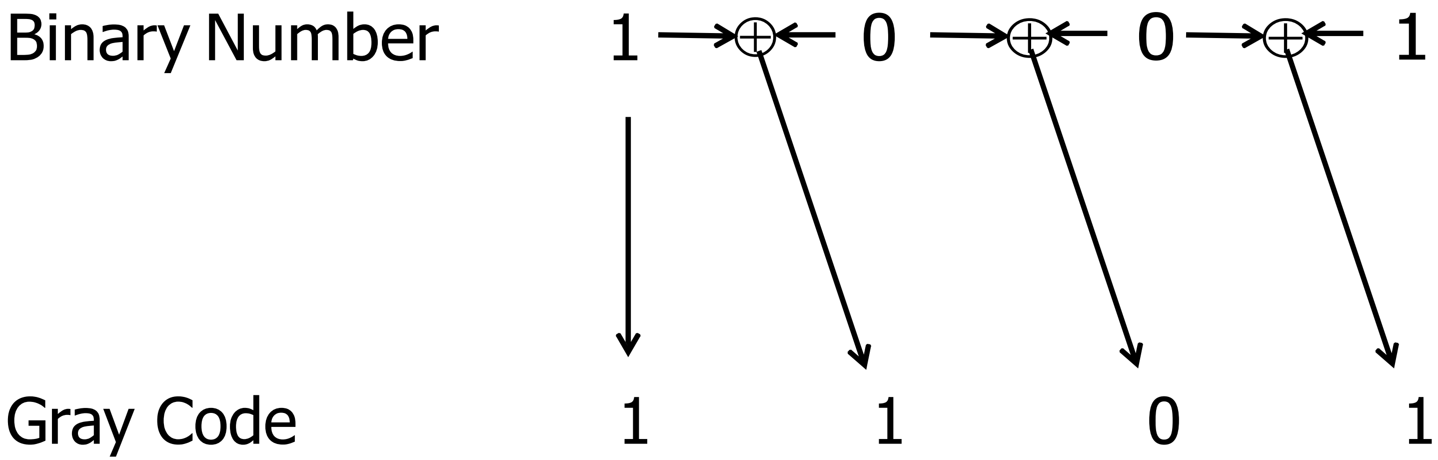
Gray Code 1 1 1 0

Binary to Gray Code

Example 2: Convert 1001 Binary Number into Gray Code

Binary to Gray Code

Example 2: Convert 1001 Binary Number into Gray Code

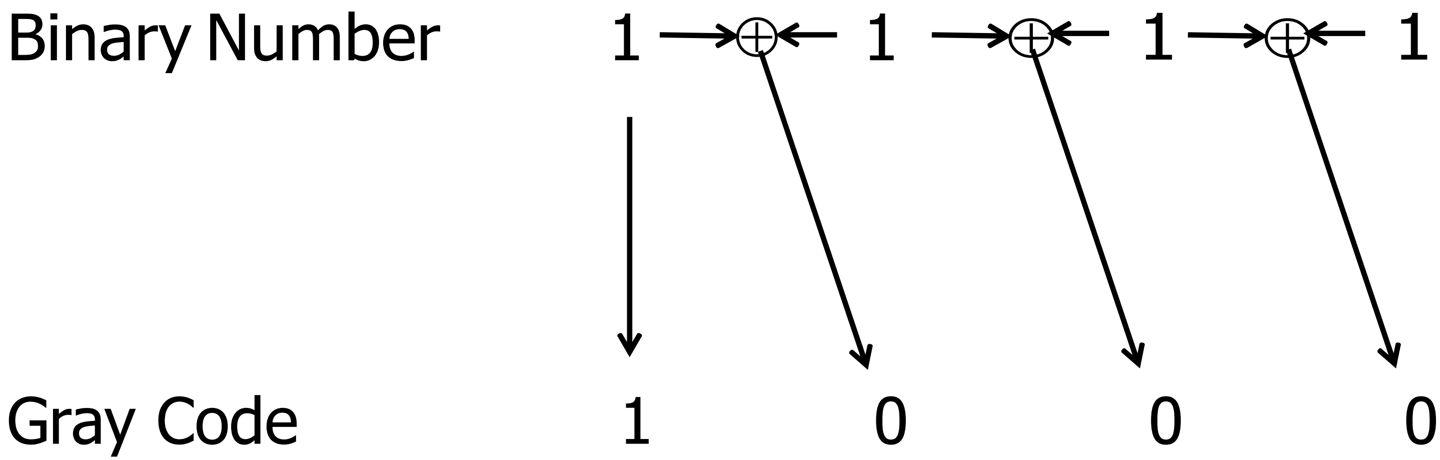


Binary to Gray Code

Example 3: Convert 1111 Binary Number into Gray Code

Binary to Gray Code

Example 3: Convert 1111 Binary Number into Gray Code



Binary to Gray Code

Example 4: Convert 1010 Binary Number into Gray Code

Binary to Gray Code

Example 4: Convert 1010 Binary Number into Gray Code

Binary Number	1	$\oplus$	0	$\oplus$	1	$\oplus$	0
Gray Code	1		1		1		1

Binary and Corresponding Gray Codes

Decimal No.	Binary No.	Gray Code
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001

15	1111	1000

Exercise

- Convert following Binary Numbers into Gray Code

    1. $(1011)_2$
    2. $(110110010)_2$
    3. $(10101010110101)_2$
    4. $(100001)_2$

Gray Code to Binary

- ✓ If an n bit gray code is represented by $G_n, G_{n-1}, ...G_1$ and its binary equivalent $B_n, B_{n-1}, ... B_1$ then binary bits are obtained from gray bits as follows;

$B_n = G_n$	$B_{n-1} = B_n \oplus G_{n-1}$	$B_{n-2} = B_{n-1} \oplus G_{n-2}$	$\cdots\cdots\cdots\cdots$	$B_1 = B_2 \oplus G_1$

*where the symbol represents Exclusive-OR operation

Gray Code to Binary

Example 1: Convert 1110 Gray code into Binary Number.

Gray Code to Binary

Example 1: Convert 1110 Gray code into Binary Number.

Gray Code	1	1	1	0

Example

Continu

Gray Code	1	1	1	0
Binary Number	1			

Example Continu

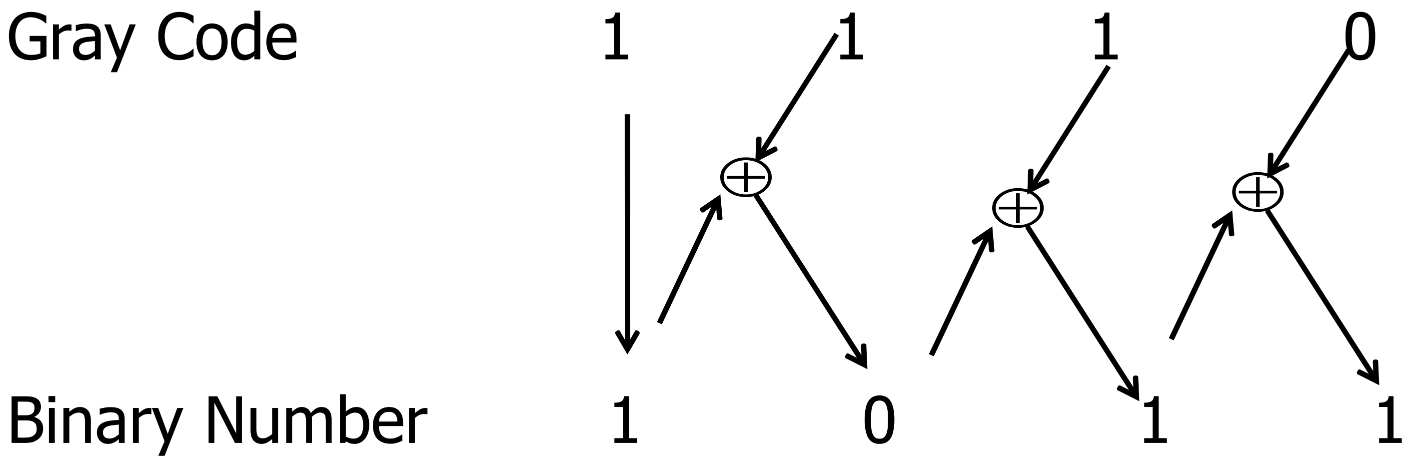
Gray Code	1	1	1	0
Binary Number	1	0		

Gray Code	1	1	1	0
Binary Number	1	0	1	

Example Continu

Gray Code	1	1	1	0
			$\oplus$	
Binary Number	1	0	1	1

Example Continu

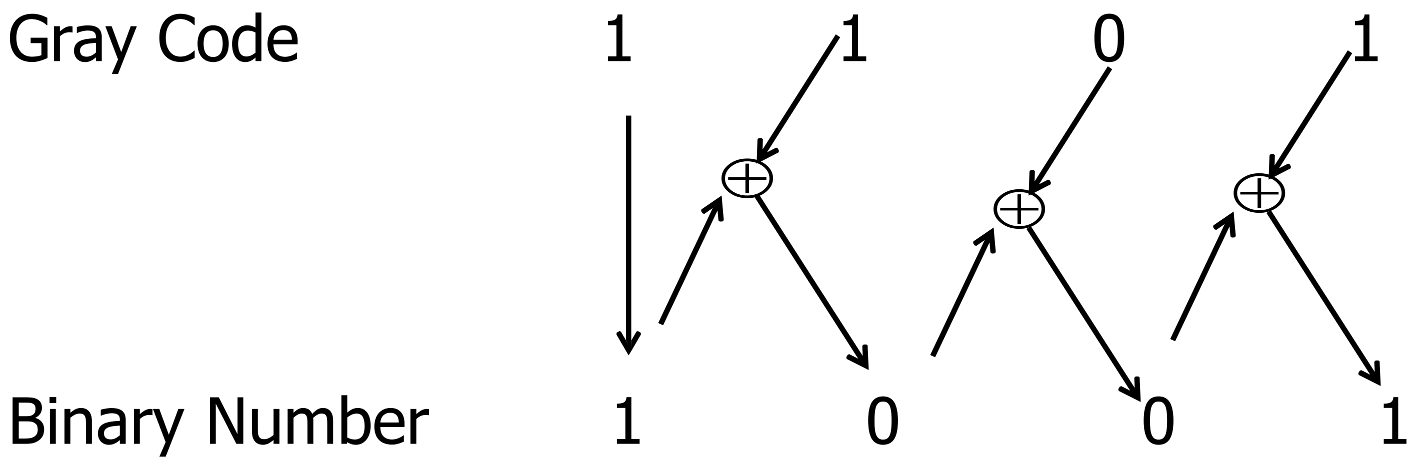


Gray Code to Binary

Example 2: Convert 1101 Gray code into Binary Number.

Gray Code to Binary

Example 2: Convert 1101 Gray code into Binary Number.

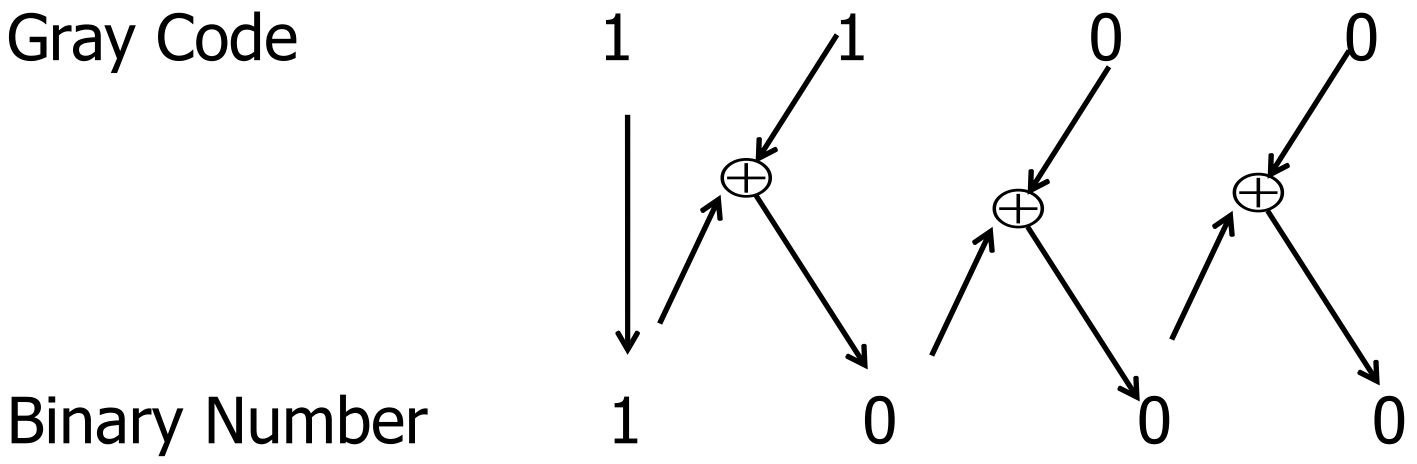


Gray Code to Binary

Example 3: Convert 1100 Gray code into Binary Number.

Gray Code to Binary

Example 3: Convert 1100 Gray code into Binary Number.



Exerci

- Convert following Gray Numbers into Binary Numbers

1. $(1111)_{GRAY}$
2. $(101110)_{GRAY}$
3. $(100010110)_{GRAY}$
4. $(11100111)_{GRAY}$

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- ✓ **Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓ Subtraction using 1's complement and 2's complement
- ✓ **Codes:** BCD, Gray Code, **Excess-3,** ASCII code
- ✓ **BCD Arithmetic:** BCD Addition

Excess-3 Code

- ✓ The Xs-3 is non-weighted BCD code.
- ✓ This code derives its name from the fact that each binary code word is the corresponding 8421 code word plus 0011.
- ✓ It is a sequential code & therefore can be used for arithmetic operations.
- ✓ It is a self complementing code

Excess-3 Code

Decimal No.	BCD Code	Excess-3 Code= BCD + Excess-3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011

Excess-3 Code

9	1001	1100

Excess-3 Code

Example 1: Obtain Xs-3 Code for 428 Decimal

Excess-3 Code

Example 1: Obtain Xs-3 Code for 428 Decimal

	4	2	8
	010	001	100
	0	0	0
+	001	001	001
	1	1	1
	011	010	101
	1	1	1

Exerci

- Convert following Decimal Numbers into Excess- 3 Code

    1. $(40)_{10}$
    2. $(88)_{10}$
    3. $(64)_{10}$
    4. $(23)_{10}$

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- ✓ **Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓ Subtraction using 1's complement and 2's complement
- ✓ **Codes:** BCD, Gray Code, Excess-3, **ASCII code**
- ✓ **BCD Arithmetic:** BCD Addition

ASCII

- ✓ The **American Standard Code for Information Interchange** is a character-encoding scheme originally based on the English alphabet.
- ✓ ASCII codes represent text in computers, communications equipment, and other devices that use text.
- ✓ Most modern character-encoding schemes are based on ASCII, though they support many

ASCII

additional characters.

ASCII

- ✓ ASCII developed from telegraphic codes. Its first commercial use was as a seven-bit tele-printer code promoted by Bell data services.
- ✓ Work on the ASCII standard began on October 6, 1960, with the first meeting of the American Standards Association's (ASA) X3.2 subcommittee.
- ✓ The first edition of the standard was published

ASCII

during 1963.

ASCII

- ✓ ASCII includes definitions for 128 characters: 33 are non-printing control characters (many now obsolete) that affect how text and space is processed and 95 printable characters, including the space (which is considered an invisible graphic)

ASCII

ASCII Codes				B7	0	0	0	0	1	1	1	1
				B6	0	0	1	1	0	0	1	1
				B5	0	1	0	1	0	1	0	1
B4	B3	B2	B1		0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	‘	p
0	0	0	1	1	SOH	DC1	|	1	A	Q	a	q
0	0	1	0	2	STX	DC2	“	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	‘	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	“	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FC	,	<	L	\	l	!
1	1	0	1	13	CR	GS		=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~

ASCII

1	1	1	1	15	SI	US	/	?	O	_	o	DEL

Unit I – Number System and Codes

- ✓ **Number System:** Base or radix of number systems, Binary, Octal, Decimal and Hexadecimal number system.
- ✓ **Binary arithmetic:** Addition, Subtraction, Multiplication, Division.
- ✓ Subtraction using 1's complement and 2's complement
- ✓ **Codes:** BCD, Gray Code, Excess-3, ASCII code
- ✓ **BCD Arithmetic: BCD Addition**

BCD

- ✓ The BCD addition is performed by individually adding the corresponding digits of the decimal number expressed in 4 bit binary groups starting from LSD.
- ✓ If there is no carry & the sum term is not an illegal code, no correction is needed.

BCD

- ✓ If there is a carry out of one group to the next group or if the sum term is an illegal code then 6 i.e. 0110 is added to the sum term of that group and resulting carry is added to the next group.

BCD

- ✓ This is done to skip the six illegal states.

BCD

Addition of two BCD numbers

Sum≤9, Carry=0

Answer is correct. No correction required.

Sum≤9, Carry=1

Add 6 to the sum term to get the correct answer

Sum>9, Carry=0

Add 6 to the sum term to get the correct answer

BCD

Example: Perform in BCD $(57)_{10} + (26)_{10}$

BCD

Example: Perform in BCD $(57)_{10} + (26)_{10}$

$$
\begin{array}{rr}
 & 57 \\
+ & 26 \\
\hline
 & 83
\end{array}
\qquad
\begin{array}{rcccccccc}
 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
+ & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
\hline
0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1
\end{array}
$$

Final Carry 0

Valid BCD Code

Invalid BCD Code

Thus we have to add 0110 in illegal BCD code

Example **Continue**

Add 0110 in only invalid code

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

$$(57)_{10} + (26)_{10} = (83)_{10}$$

Exercise

- Perform BCD Addition
- 1. $(275)_{10} + (493)_{10}$
- 2. $(109)_{10} + (778)_{10}$

3. $(88.7)_{10} + (265.8)_{10}$

4. $(204.6)_{10} + (185.56)_{10}$