# CHAPTER NO.1
# DIGITAL SYSTEM, NUMBER SYSTEM AND CONVERSION, COMPLEMENTS, SIGNED BINARY NUMBER
# DIGITAL LOGIC DESIGN

**Iqra Chaudhary (Lecturer CS dept. NUML)**

# Outline of Chapter 1

- Digital and analogue signal
- Digital Systems
- Number system
- Binary Numbers
- Octal and Hexadecimal Numbers
- Number-base Conversions
- Binary, octal and hexadecimal addition
- Binary, octal and hexadecimal multiplication
- Binary, octal and hexadecimal subtraction
- Complements
- Signed Binary Numbers

# Analogue Quantities

Continuous Quantity

- Intensity of Light
- Temperature
- Velocity

# Digital Quantities

- Discrete set of values

# Analog and Digital Signal

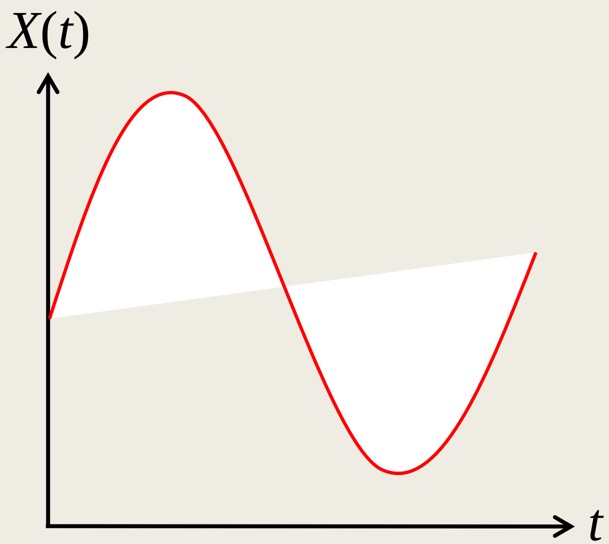
- Analog Signal

The physical quantities or signals may vary continuously over a specified range.

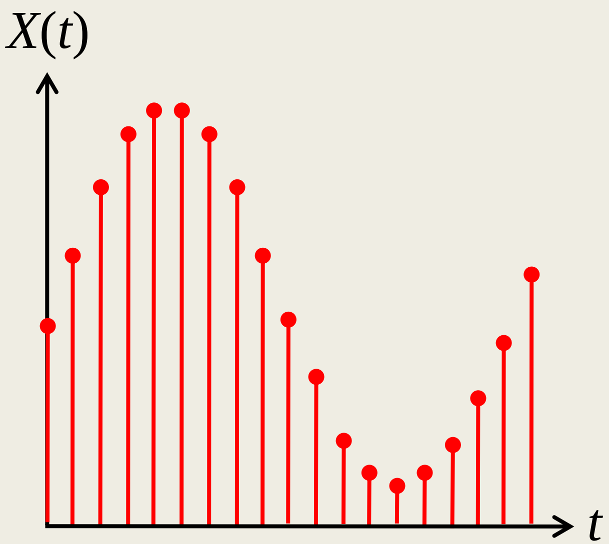
- Digital Signal

The physical quantities or signals can assume only discrete values.

  – *Greater accuracy*



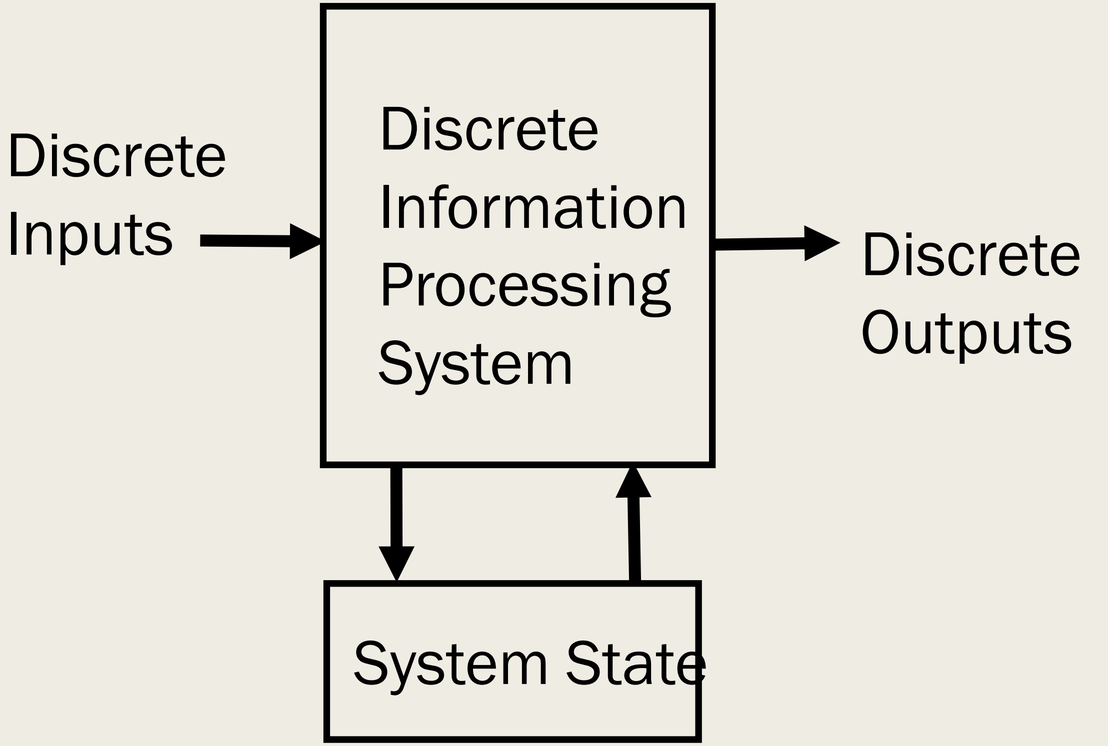

Analog signal




Digital signal

# Digital System

- **Takes a set of discrete information <u>inputs</u> and discrete internal information <u>(system state)</u> and generates a set of discrete information <u>outputs</u>.**



# Digital Systems

- Two Voltage Levels
- Two States
  - *On/Off*
  - *Black/White*
  - *Hot/Cold*
  - *Stationary/Moving*

*How does one represent more than two states in a digital system?*

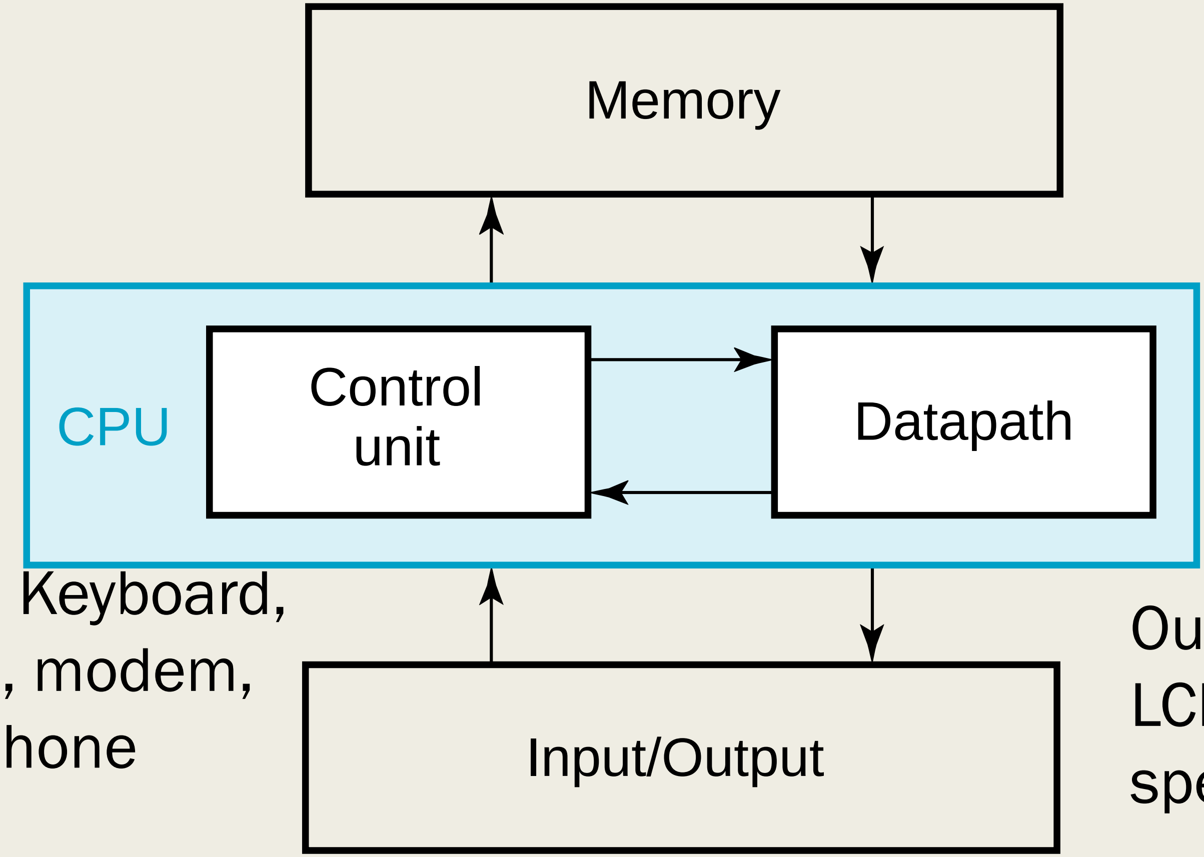
# Types of Digital Systems

- No state present
  - *Combinational Logic System*
  - *Output = Function(Input)*
- State present
  - *State updated at discrete times*

    *=> Synchronous Sequential System*
  - *State updated at any time*

    *=>Asynchronous Sequential System*
  - *State = Function (State, Input)*
  - *Output = Function (State)*
    *or Function (State, Input)*

# A Digital Computer Example




Inputs: Keyboard, mouse, modem, microphone

Outputs: CRT, LCD, modem, speakers

Synchronous or Asynchronous?

# Merits of Digital Systems

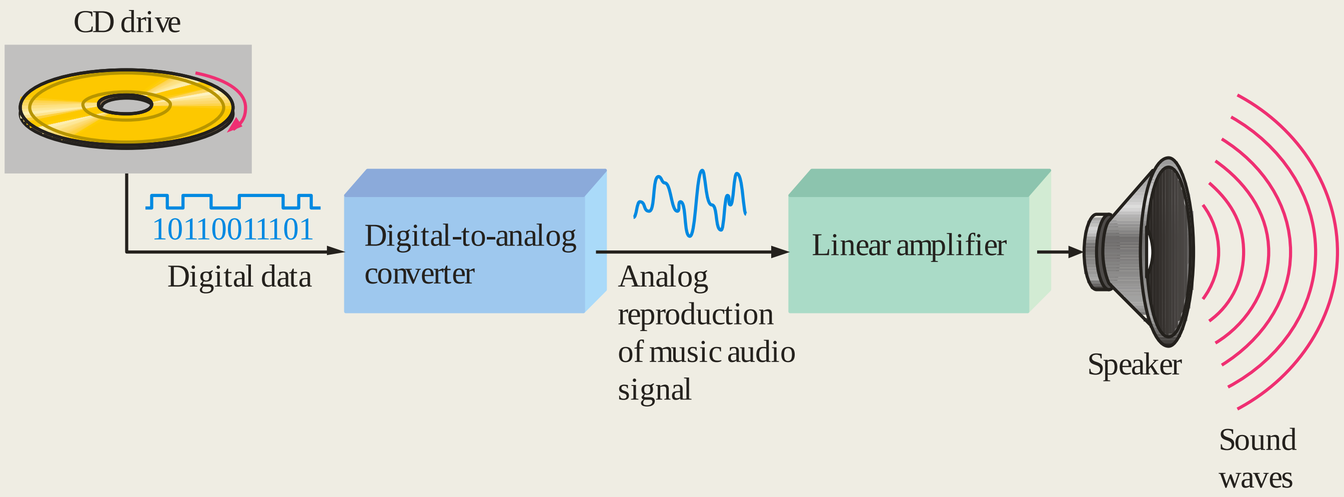
- Efficient **Processing & Data Storage**
- Efficient & Reliable **Transmission**
- **Detection and Correction of Errors**
- Precise & Accurate **Reproduction**
- Easy **Design and Implementation**
- Occupy **minimum space**

# Digital system examples

- Cell phone
- Digital camera
- MP3 / MP4 Player (IPod)
- Security systems
- Industrial process controller, etc.

# Analog and Digital Systems

- Many systems use a mix of analog and digital electronics to take advantage of each technology. A typical CD player accepts digital data from the CD drive and converts it to an analog signal for amplification.



# NUMBER SYSTEM

Decimal, Binary, octal and hexadecimal number system

# Number Systems

| | Range | Base/radix | Example | |
|---|---|---|---|---|
| Decimal | 0 ~ 9 | 10 | $(97.9)_{10}$ | 0,1,2,3,4,5,6,7,8,9 |
| Binary | 0 ~ 1 | 2 | $(110.1001)_2$ | 0,1 |
| Octal | 0 ~ 7 | 8 | $(57.32)_8$ | 0,1,2,3,4,5,6,7 |
| Hexadecimal | 0 ~ F | 16 | $(A9.10F)_{16}$ | 0,1,2,3,4,5,6,7 8,9, A,B, C, D, E, F |

# General Number System

- General form of base r: $$\sum_i a_i \times r^i$$
- Base r is also called radix
  - In decimal system r = 10;
  - In binary r = 2
- The range value of an n-digit number in radix r is
  - Minimum value: 0
  - Maximum value: $r^n$ -1
  - Number of different values: $r^n$

# Decimal Number Systems

- Common numbering system is "base 10"
  - Why?

- Numbers in base 10
  - Ten different digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - Number is represented by a sequence of digits:

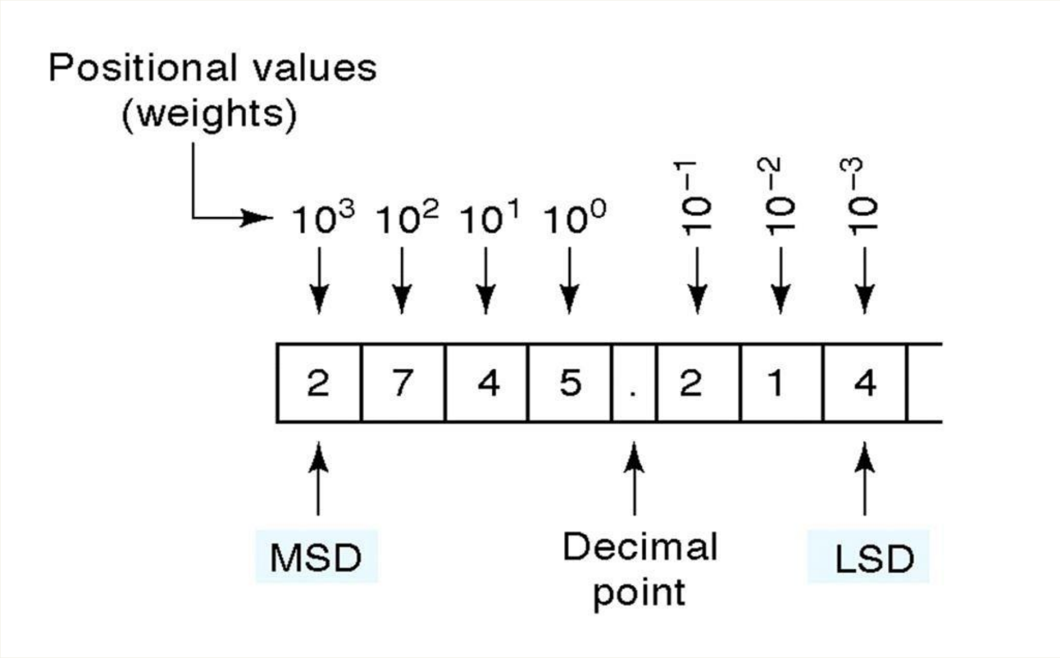
$$a_n\, a_{n-1} \ldots\ a_1 a_0$$

  - Value of a number in base 10 is

$$a_n \times 10^n + a_{n-1}\ \times\ 10^{n-1} + \ldots a_1 \times 10^1\ \times\ a_0 \times 10^0$$

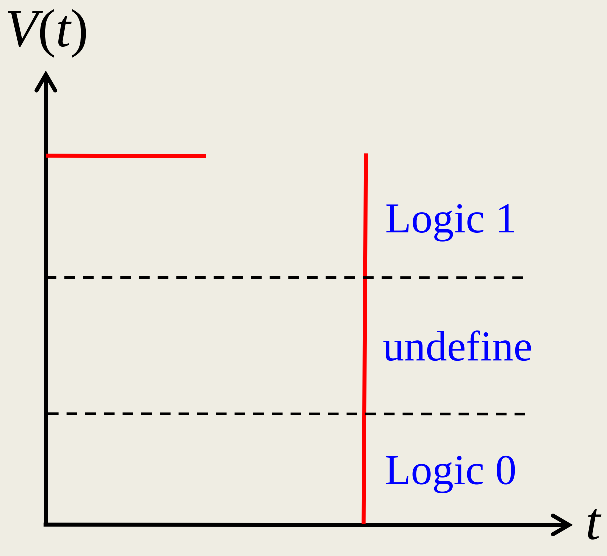
# Decimal Numbers

- In decimal number system there are ten symbols, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
- Express the decimal number 2745.214 as a sum of the values of each digit



# Binary Number System

- Binary Numbers
- Representing Multiple Values or ranges of values of physical quantities
- Combination of 0v & 5v
- Binary values are represented abstractly by:
  - *Digits 0 and 1*



Binary digital signal

# Binary numbers

- Base 2 number use only two symbols: 0, 1
  - Why ?
- Digits need to be represented in a system
  - Electronic systems typically use voltage levels
  - Representing 10 different voltages reliably is difficult
  - Binary decision is much easier (on, off)
- Binary representation is ideal
  - Minimal number of digits
  - Easily represented in voltages

# Binary Signal

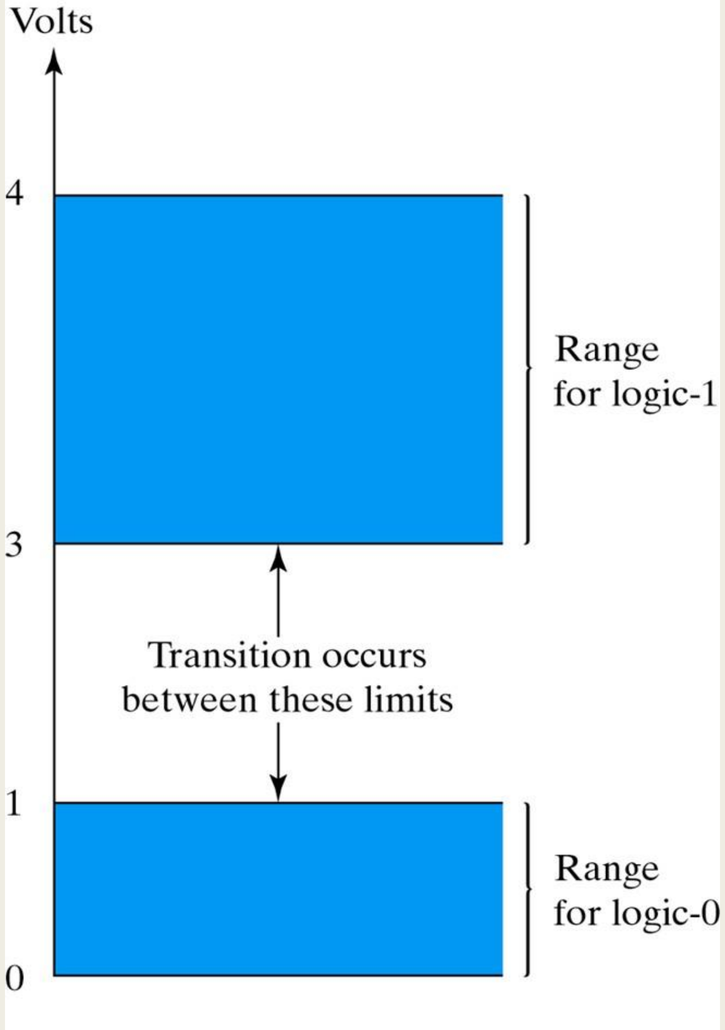



Fig. 1-3 Example of binary signals

# Examples for Binary Numbers

- Value is represented by $(01001)_2$
  - leading zero makes no difference
  - $(1001)_2$ translate into $1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 = 8+0+0+1=(9)_{10}$
- Same process for numbers with decimal point
  - what is the value of $(1001.1001)_2$ ?
  - $(1001.1001)_2 = 1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 + 2^{-1} + 0x\ 2^{-2} + 0x\ 2^{-3} + 1x\ 2^{-4} = 8+0+0+1+1/2+0+0+1/16=(9.5625)_{10}$
  - Important: it's NOT $(9.9)_{10}$!

# Binary Numbers

- Strings of binary digits (“bits”)
- One bit can store a number from 0 to 1
- *n* bits can store numbers from 0 to $2^n$

Examples: (It can store from 0 to 15 only)

- 0000
- 0001
- 1001

01010100 (What is the minimum and maximum value that could be stored in it)?

# Binary Numbers (Terminology)

- Nibble: Group of four bits is called "nibble"
  - E.g, $(1101)_2$
- Byte: Group of eight bits is called "byte"
  - E.g, $(01001101)_2$
- What is the range of values of an n-bit binary number?
  - Minimum value: 0
  - Maximum value : $2^n-1$
  - Number of different values: $2^n$

# The Power of 2

| n | $2^n$ |
|---|---|
| 0 | $2^0=1$ |
| 1 | $2^1=2$ |
| 2 | $2^2=4$ |
| 3 | $2^3=8$ |
| 4 | $2^4=16$ |
| 5 | $2^5=32$ |
| 6 | $2^6=64$ |
| 7 | $2^7=128$ |

| n | $2^n$ | |
|---|---|---|
| 8 | $2^8=256$ | |
| 9 | $2^9=512$ | |
| 10 | $2^{10}=1024$ | Kilo |
| 11 | $2^{11}=2048$ | |
| 12 | $2^{12}=4096$ | |
| 20 | $2^{20}=1M$ | Mega |
| 30 | $2^{30}=1G$ | Giga |
| 40 | $2^{40}=1T$ | Tera |

# Special Powers of 2

- $2^{10}$ (1,024) is Kilo, "K“
- $2^{20}$ (1,048, 576) is Mega, "M“
- $2^{30}$ (1,073, 741, 824) is Giga, "G“
- $2^{40}$ (1,099, 511, 627, 776 ) is Tera, “T“
- Trick to simplify estimation:
- $2^{10} = 1024 \sim 1000 = 10^3$
  - Example: $2^{32} = 4\text{x}10^9$ = 4 billion
- Prefixes:
  - Kilo ($10^3 \sim 2^{10}$),
  - Mega ($10^6 \sim 2^{20}$),
  - Giga ($10^9 \sim 2^{30}$),
  - Tera ($10^{12} \sim 2^{40}$)

# NUMBER BASE CONVERSIONS

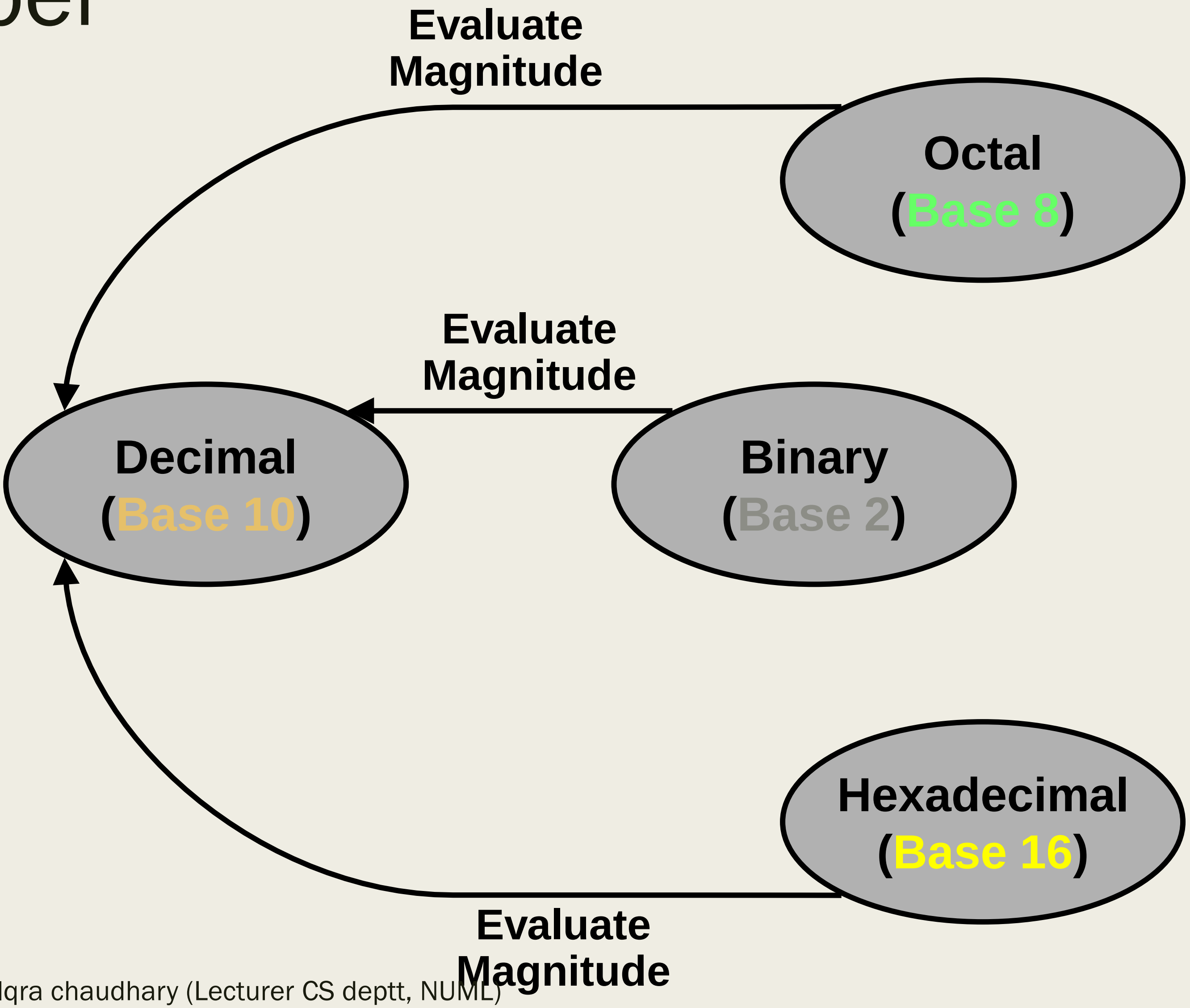
Base-r to decimal

Decimal to base-r

Octal to binary and binary to octal

Hexadecimal to binary and binary to hexadecimal

# Conversion from base r to decimal number



# Binary Number System

- Base = 2
  - *2 digits { 0, 1 }, called binary digits or "bits"*
- Evaluate magnitude

| | | | | |
|---|---|---|---|---|
| *4* | *2* | *1* | *1/2* | *1/4* |
| **1** | **0** | **1** | **0** | **1** |
| *2* | *1* | *0* | *-1* | *-2* |

$$1 * 2^2 + 0 * 2^1 + 1 * 2^0 + 0 * 2^{-1} + 1 * 2^{-2}$$

$$= (5.25)_{10}$$

$$(101.01)_2$$

- Groups of bits   4 bits = *Nibble*   **1 0 1 1**

  8 bits = *Byte*   **1 1 0 0 0 1 0 1**

# Octal Number System

- Base = 8
  - *8 digits { 0, 1, 2, 3, 4, 5, 6, 7 }*
- Evaluate magnitude

| | | | | |
|---|---|---|---|---|
| *64* | *8* | *1* | *1/8* | *1/64* |
| 5 | 1 | 2 | 7 | 4 |
| *2* | *1* | *0* | *-1* | *-2* |

$$5 * 8^{2} + 1 * 8^{1} + 2 * 8^{0} + 7 * 8^{-1} + 4 * 8^{-2}$$

$$= (330.9375)_{10}$$

$$(512.74)_{8}$$

# Hexadecimal Number System

- Base = 16
  - *16 digits { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F }*
- Evaluate magnitude

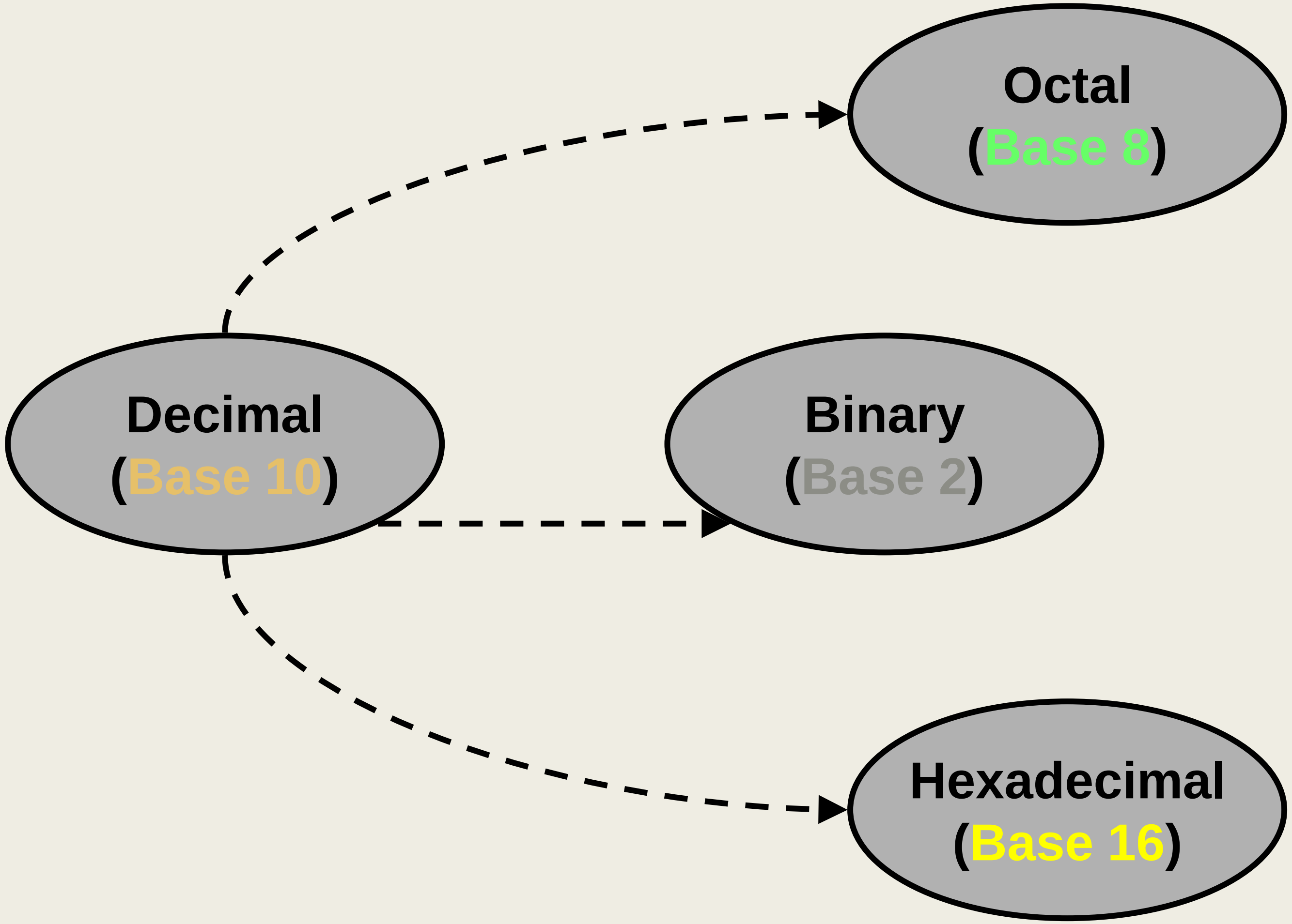
| 256 | 16 | 1 | | 1/16 | 1/256 |
|---|---|---|---|---|---|
| 1 | E | 5 | . | 7 | A |
| 2 | 1 | 0 | | -1 | -2 |

$$1 *16^{2}+14 *16^{1}+5 *16^{0}+7 *16^{-1}+10 *16^{-2}$$

$$=(485.4765625)_{10}$$

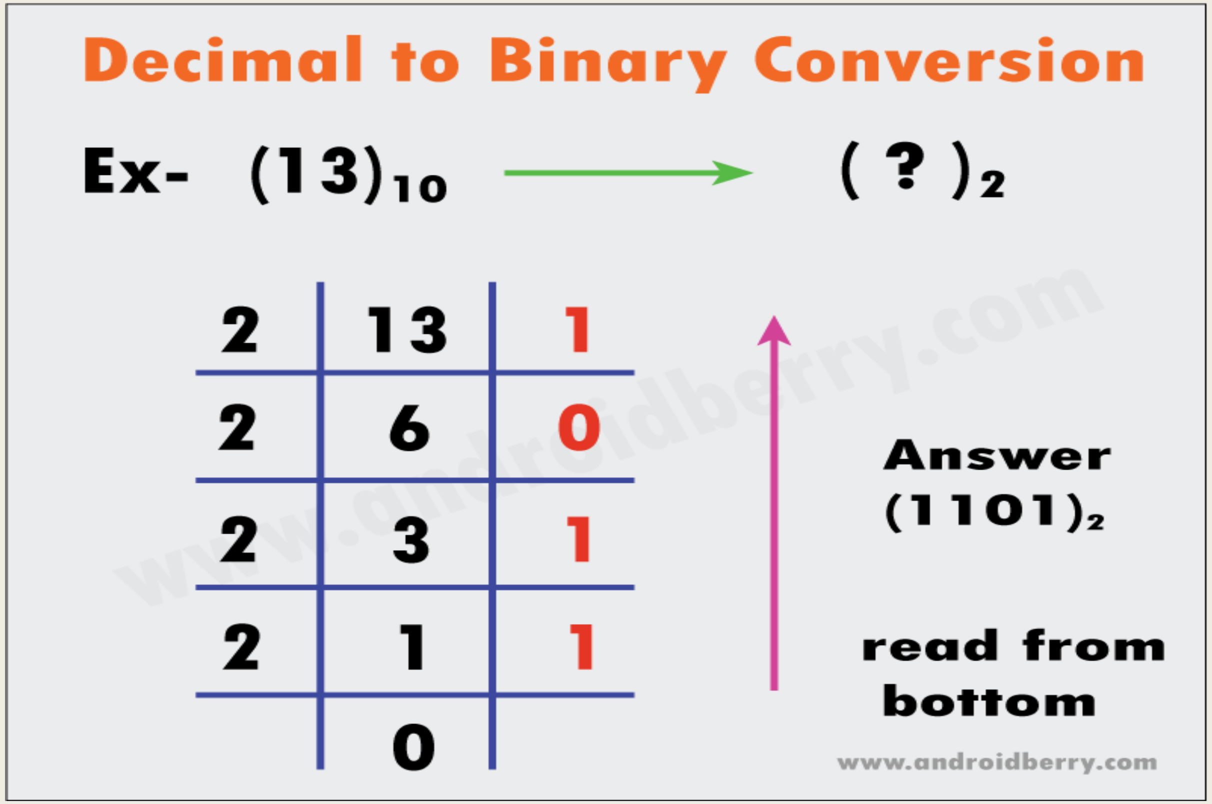
$$(1E5.7A)_{16}$$

# Conversion from decimal to base r



# Decimal (*Integer*) to Binary Conversion

Example: $(13)_{10}$




**Answer:** $(13)_{10} = (a_3\, a_2\, a_1\, a_0)_2 = (1101)_2$

MSB LSB

# Decimal (*Integer*) to Binary Conversion(SECOND METHOD)

- Divide the number by the 'Base' (=2)
- Take the remainder (either 0 or 1) as a coefficient
- Take the quotient and repeat the division

**Example: $(13)_{10}$**

| | Quotient | Remainder | Coefficient |
|---|---|---|---|
| 13 / 2 = | 6 | 1 | $a_0 = 1$ |
| 6 / 2 = | 3 | 0 | $a_1 = 0$ |
| 3 / 2 = | 1 | 1 | $a_2 = 1$ |
| 1 / 2 = | 0 | 1 | $a_3 = 1$ |

**Answer:** $(13)_{10} = (a_3\ a_2\ a_1\ a_0)_2 = (1101)_2$

**MSB** **LSB**

# Decimal (*Fraction*) to Binary Conversion

- Multiply the number by the 'Base' (=2)
- Take the integer (either 0 or 1) as a coefficient
- Take the resultant fraction and repeat the division

**Example:** $(0.625)_{10}$

| | | Integer | | Fraction | Coefficient |
|---|---|---|---|---|---|
| 0.625 | * 2 = | 1 | . | 25 | $a_{-1} = 1$ |
| 0.25 | * 2 = | 0 | . | 5 | $a_{-2} = 0$ |
| 0.5 | * 2 = | 1 | . | 0 | $a_{-3} = 1$ |

**Answer:** $(0.625)_{10} = (0.a_{-1}\ a_{-2}\ a_{-3})_2 = (0.101)_2$

MSB LSB

# Decimal to Octal Conversion

Example: $(175)_{10}$

**Answer:** $(175)_{10} = (a_2\ a_1\ a_0)_8 = (257)_8$

For example : Convert the Decimal number 175 into equivalent Octal

| 8 | 175 | Reminder |
|---|---|---|
| 8 | 21 | 7 |
| 8 | 2 | 5 |
| 8 | 0 | 2 |

Answer : $(175)_{10} = (257)_8$

Example: $(0.3125)_{10}$

| | | Integer | | Fraction | Coefficient |
|---|---|---|---|---|---|
| 0.3125 | * 8 = | 2 | . | 5 | $a_{-1} = 2$ |
| 0.5 | * 8 = | 4 | . | 0 | $a_{-2} = 4$ |

**Answer:** $(0.3125)_{10} = (0.a_{-1}\ a_{-2}\ a_{-3})_8 = (0.24)_8$

# Decimal to Octal Conversion(SCEND METHOD)

Example: $(175)_{10}$

| | Quotient | Remainder | Coefficient |
|---|---|---|---|
| 175 / 8 = | 21 | 7 | $a_0 = 7$ |
| 21 / 8 = | 2 | 5 | $a_1 = 5$ |
| 2 / 8 = | 0 | 2 | $a_2 = 2$ |

**Answer:** $(175)_{10} = (a_2\, a_1\, a_0)_8 = (257)_8$

# Decimal to hexadecimal Conversion

Example: $(175)_{10}$

$$
\begin{array}{r|l}
16 & 175 \\ \hline
 & 10 - 15
\end{array}
$$

$a_0 = A$

$a_1 = F$

**Answer:** $(175)_{10} = (a_1\ a_0)_8 = (AF)_{16}$

Example: $(0.3125)_{10}$

| | Integer | Fraction | Coefficient |
|---|---|---|---|
| $0.3125 * 16 =$ | 5 . | 0 | $a_{-1} = 5$ |

**Answer:** $(0.3125)_{10} = (0.a_{-1}\ a_{-2}\ a_{-3})_8 = (0.5)_{16}$

# Decimal to hexadecimal Conversion(SECOND METHOD)

**Example:** $(175)_{10}$

| | Quotient | Remainder | Coefficient |
|---|---|---|---|
| 175 / 16 = | 10 | 15 | $a_0$ = F |
| 10 / 16= | 0 | 10 | $a_1$ = A |

**Answer:** $(175)_{10} = (a_1\ a_0)_8 = (AF)_{16}$

# Decimal to hexadecimal Conversion

$(374.37)_{10}$

| | | |
|---|---|---|
| 16 | 374 | |
| 16 | 23 | 6 |
| 16 | 1 | 7 |
| | 0 | 1 |

$0.37 \times 16 = 5.92 = 0.92$ with Carry 5

$0.92 \times 16 = 14.72 = 0.72$ with Carry 14 (E)

$0.72 \times 16 = 11.52 = 0.52$ with Carry 11 (B)

$0.52 \times 16 = 8.32 = 0.32$ with Carry 8

$(176)_{16}$

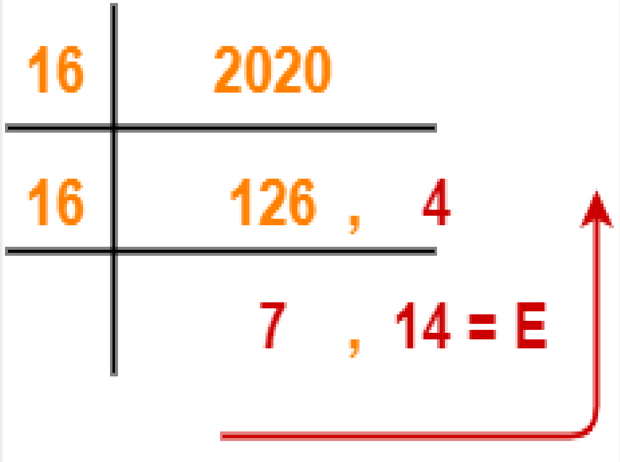
**Integer Part**

$(0.5EB8)_{16}$

**Fraction Part**

$(374.37)_{10} = (176.5EB8)_{16}$

# Decimal to hexadecimal Conversion



# Binary – Octal Conversion

- $8 = 2^3$
- Each group of 3 bits represents an octal digit

**Example:**

Assume Zeros

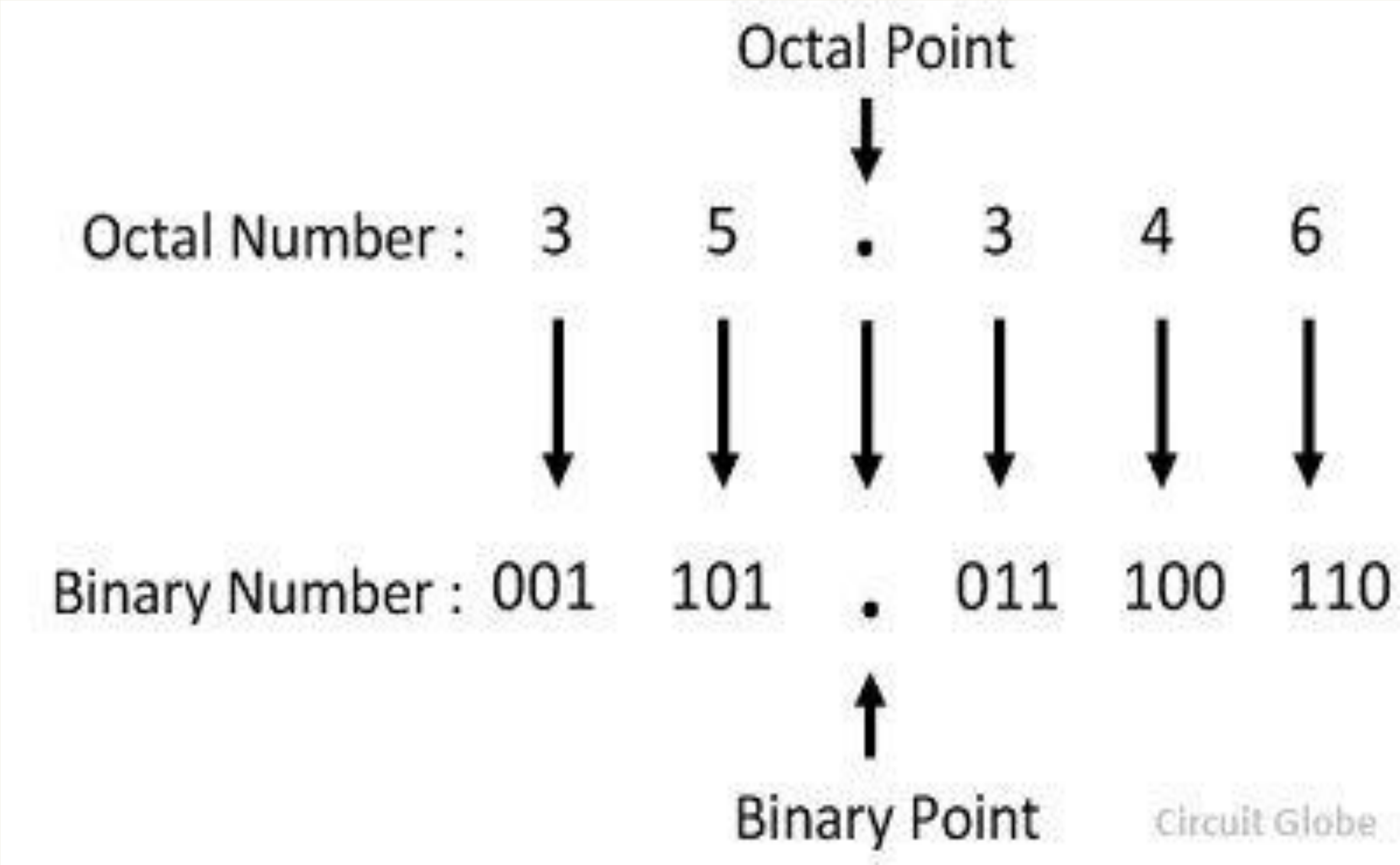
$(\ 1\,0\,1\,1\,0\,.\,0\,1\ )_2$

$(\ 2\quad 6\ .\ 2\ )_8$

| Octal | Binary |
|---|---|
| 0 | 0 0 0 |
| 1 | 0 0 1 |
| 2 | 0 1 0 |
| 3 | 0 1 1 |
| 4 | 1 0 0 |
| 5 | 1 0 1 |
| 6 | 1 1 0 |
| 7 | 1 1 1 |

**Works both ways (Binary to Octal & Octal to Binary)**

# Binary – Octal Conversion



# Binary – Hexadecimal Conversion

- $16 = 2^4$
- Each group of 4 bits represents a hexadecimal digit

**Example:**

Assume Zeros

$(\ 1\ 0\ 1\ 1\ 0\ .\ 0\ 1\ )_2$

$(\ 1\quad 6\quad .\quad 4\ )_{16}$

**Works both ways (Binary to Hex & Hex to Binary)**

| Hex | Binary |
|---|---|
| 0 | 0 0 0 0 |
| 1 | 0 0 0 1 |
| 2 | 0 0 1 0 |
| 3 | 0 0 1 1 |
| 4 | 0 1 0 0 |
| 5 | 0 1 0 1 |
| 6 | 0 1 1 0 |
| 7 | 0 1 1 1 |
| 8 | 1 0 0 0 |
| 9 | 1 0 0 1 |
| A | 1 0 1 0 |
| B | 1 0 1 1 |
| C | 1 1 0 0 |
| D | 1 1 0 1 |
| E | 1 1 1 0 |
| F | 1 1 1 1 |

# Octal – Hexadecimal Conversion

- Easier if done via binary
- ▸ 3 or 4 bit sequence correspond to digit
- Convert to Binary as an intermediate step

**Example:**

$$(2\ 6\ .\ 2)_8$$

Assume Zeros

$$(0\,1\,0\,1\,1\,0\,.\,0\,1\,0)_2$$

Assume Zeros

$$(1\ 6\ .\ 4)_{16}$$

**Works both ways (Octal to Hex & Hex to Octal)**

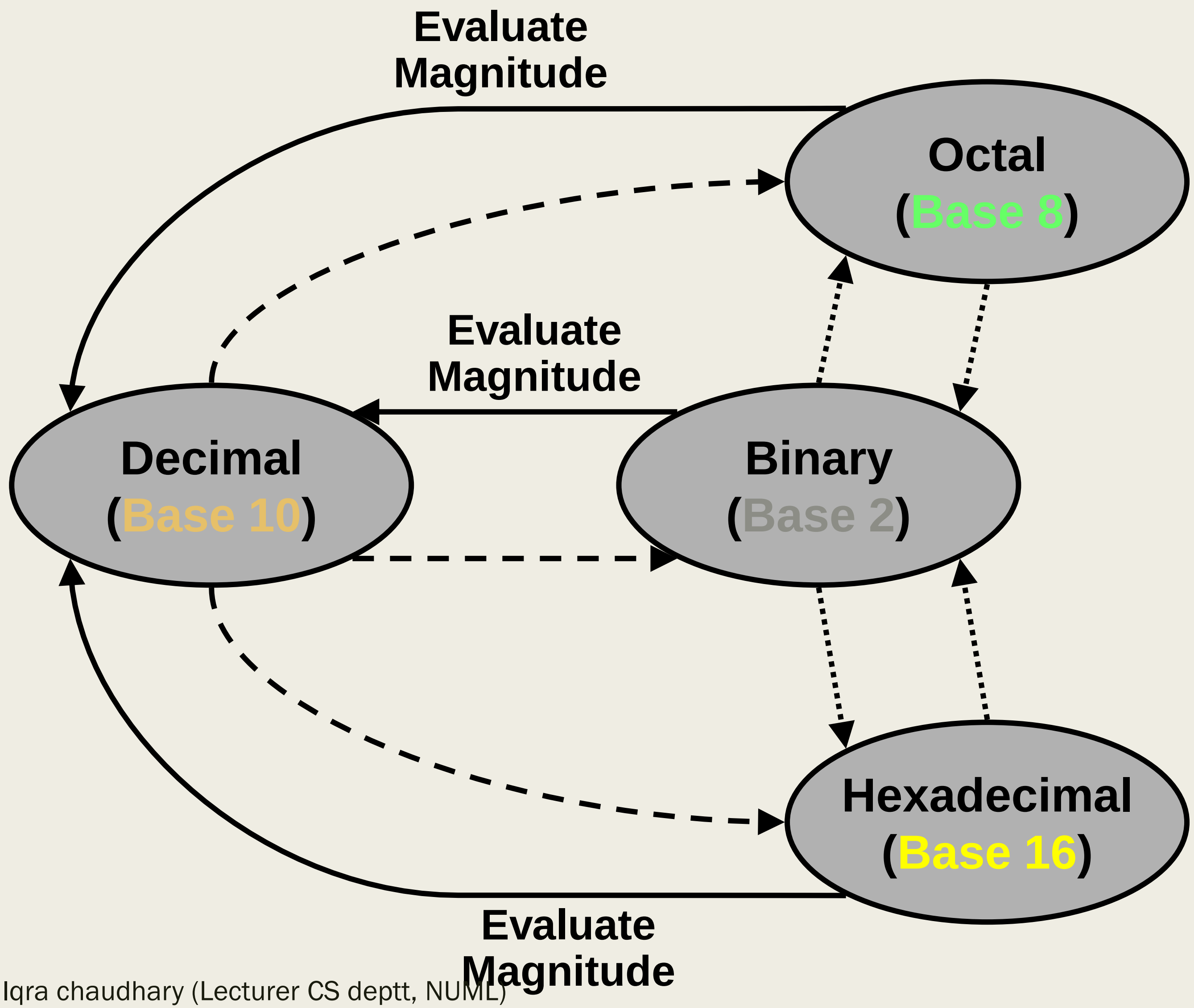
# Number Base Conversions

- Conversion to/from octal and hexadecimal
  - Example: $(4\ 5\ 5\ 6)_8$

$(2414)_{10} = (100101101110)_2$

$(9\ 6\ E)_{16}$

# Conversion


Evaluate Magnitude
Octal
(Base 8)
Evaluate Magnitude
Decimal
(Base 10)
Binary
(Base 2)
Hexadecimal
(Base 16)
Evaluate Magnitude

# Decimal, Binary, Octal and Hexadecimal

| Decimal | Binary | Octal | Hex |
|---|---|---|---|
| 00 | 0000 | 00 | 0 |
| 01 | 0001 | 01 | 1 |
| 02 | 0010 | 02 | 2 |
| 03 | 0011 | 03 | 3 |
| 04 | 0100 | 04 | 4 |
| 05 | 0101 | 05 | 5 |
| 06 | 0110 | 06 | 6 |
| 07 | 0111 | 07 | 7 |
| 08 | 1000 | 10 | 8 |
| 09 | 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 |
| 16 | 10000 | 20 | 10 |

# NUMBER SYSTEM ARITHEMETIC

Addition

# Decimal Addition

| | | | |
|---|---|---|---|
| | 1 | 1 | |
| | | 5 | 5 |
| + | | 5 | 5 |
| | 1 | 1 | 0 |

Carry

= Ten ≥ Base

➔ Subtract a Base/ Convert into required base

# Binary Addition

- Column Addition

$$
\begin{array}{cccccccc}
 & 1 & 1 & 1 & 1 & 1 & 1 & \\
 & & 1 & 1 & 1 & 1 & 0 & 1 \\
+ & & & 1 & 0 & 1 & 1 & 1 \\
\hline
 & 1 & 0 & 1 & 0 & 1 & 0 & 0
\end{array}
\qquad
\begin{array}{l}
\\
= 61 \\
= 23 \\
\\
= 84
\end{array}
$$

$\geq (2)_{10}$

# Octal Addition

Carry ⟹ 1

| | |
|---|---|
| 7 6 0 2 | 2+1=3d 3O |
| + 5 7 7 1 | 0+7=7d 7O |
| 15 5 7 3 | 6+7=13d 15O |
| | 1+7+5=13d 15O |

Carry ⟹ 1

| | |
|---|---|
| 6 5 2 | 7+5=12d 14O |
| + 5 7 4 | 1+6+5=12d 14O |
| 14 4 6 | |

# Hexadecimal Addition

## Addition in Hex.

**Example** Add the following hexadecimal number :

$$\begin{array}{r} {}^{1\,1}\phantom{F3}\\ 1AF3_{16} \\ +\ \underline{67E8_{16}} \\ 82DB_{16} \end{array} \qquad \begin{array}{r} 123_{16} \\ +\ \underline{924_{16}} \\ A47_{16} \end{array} \qquad \begin{array}{r} 90C8_{16} \\ +\ \underline{2B35_{16}} \\ BBFD_{16} \end{array}$$

72

# NUMBER SYSTEM ARITHEMETIC

Multiplication

# Binary Multiplication

- Bit by bit

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | | 1 | 0 | 1 | 1 | 1 |
| x | | | | | 1 | 0 | 1 | 0 |
| | | | | 0 | 0 | 0 | 0 | 0 |
| | | | 1 | 0 | 1 | 1 | 1 | |
| | | 0 | 0 | 0 | 0 | 0 | | |
| | 1 | 0 | 1 | 1 | 1 | | | |
| | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

# Octal Multiplication

## Octal Multiplication

```
Multiplicand:     762
Multiplier:   x    45
              -------
                 4672
                3710 -
              -------
Product:        43772
```

| Octal | Decimal | Octal |
|---|---|---|
| 5x2 = | 10= 8+2 | 12 |
| 5x6 +1= | 31= 24+7 | 37 |
| 5x7 +3= | 38= 32+6 | 46 |
| 4x2= | 8 = 8+0 | 10 |
| 4x6 +1= | 25= 24+1 | 31 |
| 4x7 +3= | 31= 24+7 | 37 |

*Form a table to calculate sums and products of 2 digits in base-r (in this case, base-8)*

# Hexadecimal multiplication

*hex*

```
     11D7
*     16E
---------
     F9C2
    6B0A
   11D7
---------
   198162
```

# NUMBER SYSTEM ARITHEMETIC

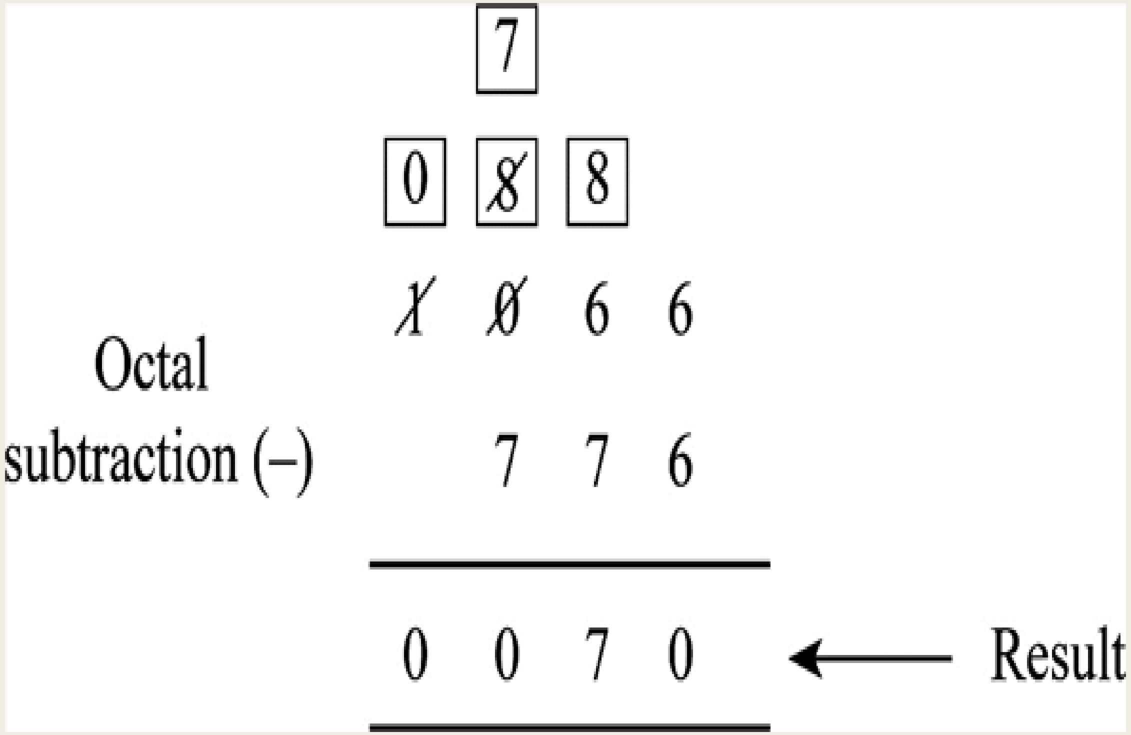
Subtraction

# Binary subtraction

**Example** Subtract $1001010_2$ - $10100_2$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | 0 | 1 | 10 | 0 | 10 | 2 | |
| | ~~1~~ | ~~0~~ | 0 | ~~1~~ | 0 | 1 | 0 |
| (-) | | | 1 | 0 | 1 | 0 | 0 |
| | | 1 | 1 | 0 | 1 | 1 | 0 |

# Octal subtraction



# Hexadecimal subtraction

The problem:

You have to subtract these numbers:
569D is Minuend
FDA is Subtrahend

$$\begin{array}{ccccc} & 5 & 6 & 9 & D \\ - & & F & D & A \\ \hline \end{array}$$

Carry Over:

1. Decimal of D is 13 and A is 10
2. (13 – 10) = 3

$$\begin{array}{ccccc} & 5 & 6 & 9 & D \\ - & & F & D & A \\ \hline & & & & 3 \end{array}$$

Carry Over:

1. 9 is smaller than D (13)
2. 9 borrow 1 from 6 so (9+16=25), 6 become 5
3. (25 - 13) = 12
4. 12 is a decimal of C

$$\begin{array}{ccccc} & & 5 & 25 & \\ & 5 & \cancel{6} & \cancel{9} & D \\ - & & F & D & A \\ \hline & & & C & 3 \end{array}$$

Carry Over:

1. 5 is smaller than F (15)
2. 5 borrow 1 from 5 so (5+16=21), 5 become 4
3. (21 – 15) = 6

$$\begin{array}{ccccc} & 4 & 21 & & \\ & & \cancel{5} & & \\ & \cancel{5} & \cancel{6} & 9 & D \\ - & & F & D & A \\ \hline & & 6 & C & 3 \end{array}$$

Carry Over:

1. (4+0) = 4

$$\begin{array}{ccccc} & 4 & & & \\ & \cancel{5} & 6 & 9 & D \\ - & 0 & F & D & A \\ \hline & 4 & 6 & C & 3 \end{array}$$

# Hexadecimal subtraction

$$
\begin{array}{cccccc}
 & & \mathrm{E} & 21 & 6 & 26 \\
 & 3 & \cancel{\mathrm{F}} & \cancel{5} & \cancel{7} & \cancel{\mathrm{A}}_{16} \\
- & & \mathrm{C} & 8 & 5 & \mathrm{E}_{16} \\
\hline
 & 3 & 2 & \mathrm{D} & 1 & \mathrm{B}_{16} \\
\hline
\end{array}
$$

# Complements

There are two types of complements for each base-*r* system:

- Radix complement and
- Diminished radix complement.

# Diminished Radix Complement OR (r-1)'s Complement

- Diminished Radix Complement OR (r-1)'s Complement
  - *Given a number N in base r having n digits, the (r–1)'s complement of N is defined as:*

$$(r^n - 1) - N$$

- **Example:**Base-10 (**for 6-digit <u>decimal</u> numbers)**
  - *9's complement is* $(r^n - 1) - N = (10^6 - 1) - N = 999999 - N$
  - *9's complement of 546700 is 999999–546700 = 453299*
- **Example:**Example: Base-2 (**for 7-digit <u>binary</u> numbers)**
  - *1's complement is* $(r^n - 1) - N = (2^7 - 1) - N = 1111111 - N$
  - *1's complement of 1011000 is 1111111–1011000 = 0100111*
- **Observation:**
  - *Subtraction from* $(r^n - 1)$ *will never require a borrow*
  - *Diminished radix complement can be computed digit-by-digit*
  - *For binary: 1 – 0 = 1 and 1 – 1 = 0*

# Diminished Radix Complement OR (r-1)'s **Complement** for base 2

- 1's Complement (*Diminished Radix* Complement)
  - *All '0's become '1's*
  - *All '1's become '0's*

  *Example* $(10110000)_2$

  ⇨ $(01001111)_2$

  *If you add a number and its 1's complement …*

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

# Radix Complement OR (r)'s Complement

- Radix Complement
  - The $r$'s complement of an $n$-digit number $N$ in base $r$ is defined as $r^n - N$ for $N \neq 0$ and as 0 for $N = 0$.
  - Comparing with the $(r - 1)$ 's complement, we note that the $r$'s complement is obtained by adding 1 to the $(r - 1)$ 's complement, since $r^n - N = [(r^n - 1) - N] + 1$.
- Example: Base-10

  The 10's complement of 012398 is 987602
  The 10's complement of 246700 is 753300

- Example: Base-2

  The 2's complement of 1101100 is 0010100
  The 2's complement of 0110111 is 1001001

# Radix Complement OR (r)'s Complement

- 2's Complement (*Radix* Complement)
  - *Take 1's complement then add 1*

| | | | |
|---|---|---|---|
| *Example:* | 1 0 1 1 0 0 0 0 | OR | 1 0 1 1 0 0 0 0 |
| *Number:* | | | |
| *1's Comp.:* | 0 1 0 0 1 1 1 1 | | |
| | + 1 | | |
| | 0 1 0 1 0 0 0 0 | | 0 1 0 1 0 0 0 0 |

# Complements

- Subtraction with Complements
  - *The subtraction of two n-digit unsigned numbers M – N in base r can be done as follows:*

1. Add the minuend $M$ to the $r$'s complement of the subtrahend $N$. Mathematically, $M + (r^n - N) = M - N + r^n$.
2. If $M \geqq N$, the sum will produce and end carry $r^n$, which can be discarded; what is left is the result $M - N$.
3. If $M < N$, the sum does not produce an end carry and is equal to $r^n - (N - M)$, which is the $r$'s complement of $(N - M)$. To obtain the answer in a familiar form, take the $r$'s complement of the sum and place a negative sign in front.

# Complements: Subtraction using radix complement

- Example 1.5
  - *Using 10's complement, subtract 72532 – 3250.*

$$\begin{array}{rr} M = & 72532 \\ \text{10's complement of} \quad N = & \underline{+96750} \\ \text{Sum} = & 169282 \\ \text{Discard end carry } 10^5 = & \underline{-100000} \\ \text{Answer} = & 69282 \end{array}$$

- Example 1.6
  - *Using 10's complement, subtract 3250 – 72532.*

$$\begin{array}{rr} M = & 03250 \\ \text{10's complement of} \quad N = & \underline{+27468} \\ \text{Sum} = & 30718 \end{array}$$

There is no end carry.

Therefore, the answer is – (10's complement of 30718) = – 69282.

# Complements: Subtraction using radix complement

- Example 1.7
  - *Given the two binary numbers X = 1010100 and Y = 1000011, perform the subtraction (a) X – Y ; and (b) Y – X, by using 2's complement.*

(a)

| | |
|---|---|
| X = | 1010100 |
| 2's complement of Y = | +0111101 |
| Sum = | 10010001 |
| Discard end carry $2^7$ = | – 10000000 |
| Answer. X – Y = | 0010001 |

(b)

| | |
|---|---|
| Y = | 1000011 |
| 2's complement of X = | + 0101100 |
| Sum = | 1101111 |

There is no end carry. Therefore, the answer is Y – X = – (2's complement of 1101111) = – 0010001.

# Complements: Subtraction using diminished radix complement

- Subtraction of unsigned numbers can also be done by means of the ($r$ – 1)'s complement. Remember that the ($r$ – 1) 's complement is one less then the $r$'s complement.
- Example 1.8
  - *Repeat Example 1.7, but this time using 1's complement.*

(a) $X - Y = 1010100 - 1000011$

$$\begin{array}{rr}
X = & 1010100 \\
\text{1's complement of } Y = & \pm\, 0111100 \\
\text{Sum} = & 10010000 \\
\text{End-around carry} = & \underline{+ \qquad 1} \\
\text{Answer. } X - Y = & 0010001
\end{array}$$

(b) $Y - X = 1000011 - 1010100$

$$\begin{array}{rr}
Y = & 1000011 \\
\text{1's complement of } X = & \underline{+\,0101011} \\
\text{Sum} = & 1101110
\end{array}$$

There is no end carry, Therefore, the answer is Y – X = – (1's complement of 1101110) = – 0010001.

# UnSigned and signed binary number

| Representation | Range |
|---|---|
| Signed n-bit integer | $-2^{n-1}$ to $2^{n-1}-1$ |
| Unsigned n-bit integer | $0$ to $2^{n}-1$ |
| Signed 32-bit integer | $-2^{31}$ to $2^{31}-1$ |
| Unsigned 32-bit integer | $0$ to $2^{32}-1$ |

# Signed Binary Numbers

- To represent negative integers, we need a notation for negative values.
- It is customary to represent the sign with a bit placed in the leftmost position of the number since binary digits.
- The convention is to make the sign bit 0 for positive and 1 for negative.
- Example:

| | |
|---|---|
| Signed-magnitude representation: | 10001001 |
| Signed-1's-complement representation: | 11110110 |
| Signed-2's-complement representation: | 11110111 |

- Table 1.3 lists all possible four-bit signed binary numbers in the three representations.

# Signed Binary Numbers

**Table 1.3**
*Signed Binary Numbers*

| Decimal | Signed-2's Complement | Signed-1's Complement | Signed Magnitude |
|---|---|---|---|
| +7 | 0111 | 0111 | 0111 |
| +6 | 0110 | 0110 | 0110 |
| +5 | 0101 | 0101 | 0101 |
| +4 | 0100 | 0100 | 0100 |
| +3 | 0011 | 0011 | 0011 |
| +2 | 0010 | 0010 | 0010 |
| +1 | 0001 | 0001 | 0001 |
| +0 | 0000 | 0000 | 0000 |
| −0 | — | 1111 | 1000 |
| −1 | 1111 | 1110 | 1001 |
| −2 | 1110 | 1101 | 1010 |
| −3 | 1101 | 1100 | 1011 |
| −4 | 1100 | 1011 | 1100 |
| −5 | 1011 | 1010 | 1101 |
| −6 | 1010 | 1001 | 1110 |
| −7 | 1001 | 1000 | 1111 |
| −8 | 1000 | — | — |

# Signed Binary Numbers

- Arithmetic addition
  - *The addition of two numbers in the signed-magnitude system follows the rules of ordinary arithmetic.* ***If the signs are the same****, we add the two magnitudes and give the sum the common sign.* ***If the signs are different****, we subtract the smaller magnitude from the larger and give the difference the sign if the larger magnitude.*
  - *The addition of two signed binary numbers with negative numbers represented in signed-2's-complement form is obtained from the addition of the two numbers, including their sign bits.*
  - *A carry out of the sign-bit position is discarded.*
- Example:

| | | | |
|---|---|---|---|
| + 6 | 00000110 | − 6 | 11111010 |
| +13 | 00001101 | +13 | 00001101 |
| + 19 | 00010011 | + 7 | 00000111 |
| + 6 | 00000110 | − 6 | 11111010 |
| −13 | 11110011 | −13 | 11110011 |
| − 7 | 11111001 | − 19 | 11101101 |

# Signed Binary Numbers

- Arithmetic Subtraction
  - *In 2's-complement form:*

1. Take the 2's complement of the subtrahend (including the sign bit) and add it to the minuend (including sign bit).
2. A carry out of sign-bit position is discarded.

$$(\pm A) - (+B) = (\pm A) + (-B)$$
$$(\pm A) - (-B) = (\pm A) + (+B)$$

- Example:

(– 6) – (– 13) → (11111010 – 11110011)

→ (11111010 + 00001101)

→ 00000111 (+ 7)