Digital logic design

Number systems

Dr. E. Paul Braineard

Number systems

- Outline
  - Digital systems
  - Number systems
    - Decimal
    - Binary
    - Octal
    - Hexadecimal
  - Number conversion

Dr. E. Paul Braineard

Digital systems: Definition

- Digital systems (circuits) store, process, and communicate information in **digital form (0 and 1)**, **called as 'bit' (binary digit)**
  - **Binary number system (0 and 1)**
  - Example: 10101
- Real world processing
  - **Decimal number system (0 – 9)**
- Decimal number (Real numbers)
  - Defined as a number whose **whole part** and the **fractional part** are separated by a decimal point. The dot (.) in the decimal number is called a decimal point
  - Example: **45.26**
    - (how to read? Forty five point two six **not** forty five point twenty six)

Number systems

A number is represented by a string of digits

• Decimal	(0 – 9)	[10 digits]
• Binary	(0 – 1)	[2 digits]
• Octal	(0 – 7)	[8 digits]
• Hexadecimal	(0 – 9 & A – F)	[16 digits]

Dr. E. Paul Braineard

Number systems

Number system	Symbols (Digits)	Base or Radix	Example
Decimal	0,1,2,3,4,5,6,7,8,9	10	9874.231
Binary	0,1	2	110101.0011
Octal	0,1,2,3,4,5,6,7	8	765.342
Hexadecimal	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F	16	7AG9.B64

Dr. E. Paul Braineard

Numbers: 0 to 15

Decimal (0 – 9)
Binary (0 – 1)
Octal (0 – 7)
Hexadecimal (0 – 9 & A – F)

Base or *Radix*

Decimal	Binary	Octal	Hexadecimal
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
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

Dr. E. Paul Braineard

Decimal number systems

- Base – 10 number system (0 – 9)
- Ex: $(2589)_{10}$
- a number with a decimal point is represented by a series of coefficients
  $a_5 a_4 a_3 a_2 a_1 a_0 . a_{-1} a_{-2} a_{-3} \Rightarrow 543210.123$
- The coefficients a_j are decimal digits from 0 – 9
- Subscript 'j' gives the place value
- Each digit position has an associated *weight*
- ∴The value of a number is the weighted sum of digits
- Example
  - $2589 = 2000 + 500 + 80 + 9 = 2 \times 1000 + 5 \times 100 + 8 \times 10 + 9 \times 1$
  - $2589 = 2 \times 10^3 + 5 \times 10^2 + 8 \times 10^1 + 9 \times 10^0$
- Each weight is a power of 10 corresponding to the digit's position

Dr. E. Paul Braineard

Decimal number with decimal digit

- Example 2589.36

$$= 2000 + 500 + 80 + 9 + 0.3 + 0.06$$
$$= 2 \times 1000 + 5 \times 100 + 8 \times 10 + 9 \times 1 + 3 \times 0.1 + 6 \times 0.01$$
$$= 2 \times 10^3 + 5 \times 10^2 + 8 \times 10^1 + 9 \times 10^0 + 3 \times 10^{-1} + 6 \times 10^{-2}$$

- Each weight is a power of 10 corresponding to the digit's position

Decimal number systems

- Ex: a number A of the form $a_1a_0.a_{-1}a_{-2}$ (89.36)
- has the value $A = a_1 \cdot 10^1 + a_0 \cdot 10^0 + a_{-1} \cdot 10^{-1} + a_{-2} \cdot 10^{-2}$
  ($8 \times 10^1 + 9 \times 10^0 + 3 \times 10^{-1} + 6 \times 10^{-2}$)
- where, $a_1 = 8$, $a_0 = 9$, $a_{-1} = 3$, $a_{-2} = 6$

10 is called the *base* or *radix*

10 because it uses ten digits

The decimal point is called as *radix point*

Dr. E. Paul Braineard

Unit of Memory

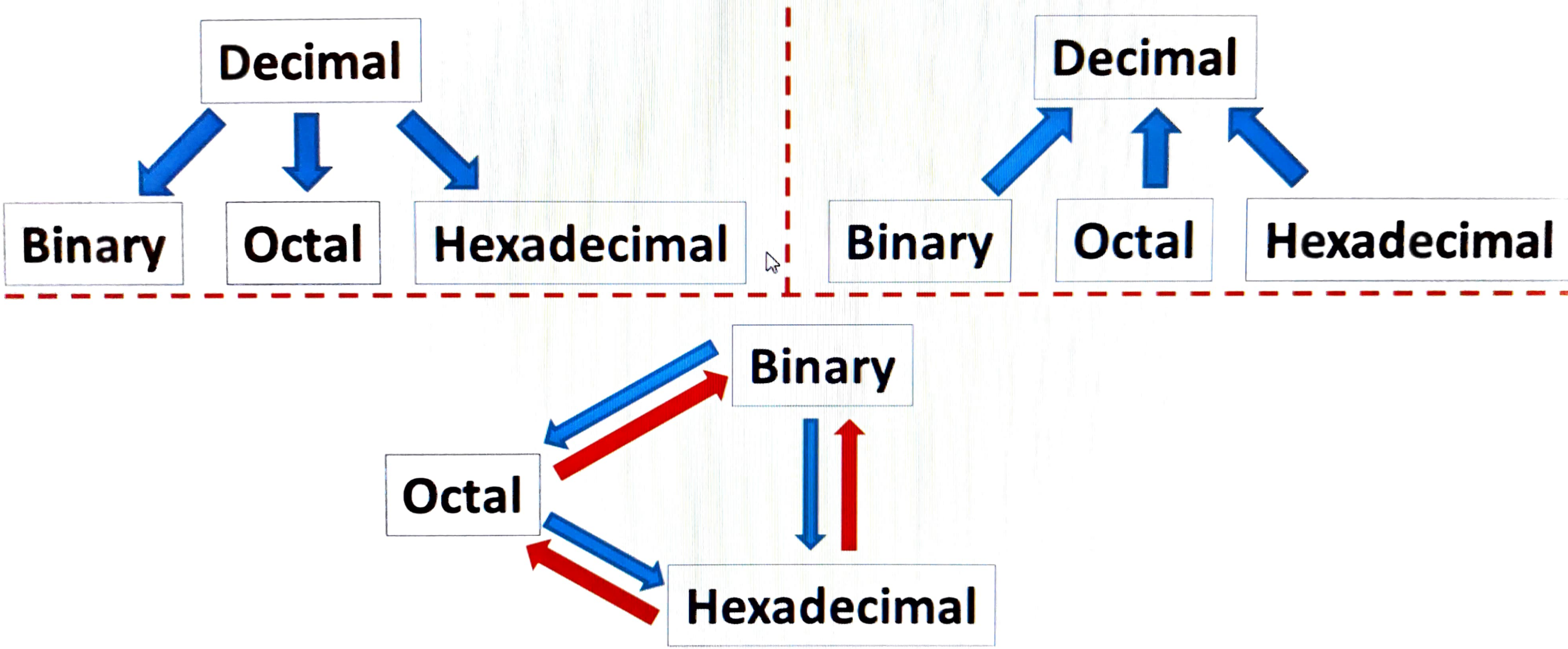
- The various units used to measure computer memory, are as follows
- **Bit**
  - Abbreviation for **b**inary dig**it**, is basic unit of memory.
  - It is smallest unit of information.
  - Bit is represented by a lower case '**b**'.
- **Byte**
  - A unit of **8 bits** is known as a byte. Hence, a byte is able to contain any binary number between 00000000 and 11111111.
  - Byte is represented by uppercase '**B**'.

Memory representation

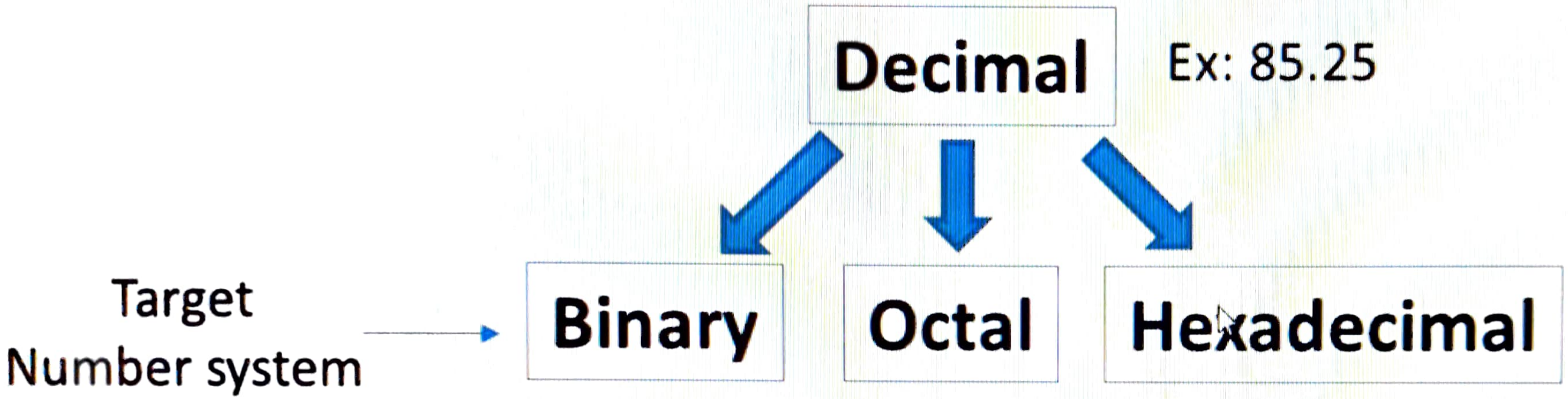
Kilobyte	KB	1024 B
Megabyte	MB	1024 KB
Gigabyte	GB	1024 MB
Terabyte	TB	1024 GB

B = bytes

Number base conversions



Number base conversions



- **Divide integer part of the decimal number by the target base or radix**
- **Multiply the fractional part of the decimal number by the target base or radix**
- r is 2, 8, 16 respectively
- r = base or radix of the system to which the number is to be converted, target number system

Number base conversions

Conversion involves two steps (Ex: 31.6875)

- Integer part
  - Divide the decimal number by the base (r) of the target number system
  - Reminder is LSD/LSB
    - Continue dividing till quotient is zero
  - Reminder is MSD/MSB
- Fractional part
  - Multiply the fractional part by base (r) of the target number system
  - the integral part is MSD/MSB
    - Continue multiplying by r until the fractional part is 0
  - The integral part is LSD /LSB
  - If a fractional part repeats stop, the digits recur

Decimal (31.6875) to Binary Conversion

- **Integer Conversion**
  - Divide the decimal number by '2, until the quotient is 0.
  - Write the remainders in order.
  - The first remainder gives the lowest order bit, **Least significant bit**
  - The last remainder gives the highest order bit, **Most significant bit**
  - Ex: Convert $(31)_{10}$ to binary

31

eting details

Raise hand

Turn on captions

Paul Braineard is presenting

Decimal (31.6875) to Binary Conversion

Fraction Conversion

- Multiply by ‘2’ till the fraction disappears
- Note down the integer part after every multiplication
- Write down from first till last integer part
- Ex: Convert $(0.6875)_{10}$ to binary number

	Product	Integer + Fraction	Coefficient
0.6875 × 2	1.3750	1 + 0.3750	$a_{-1} = 1$
0.3750 × 2	0.7500	0 + 0.7500	$a_{-2} = 0$
0.7500 × 2	1.5000	1 + 0.5000	$a_{-3} = 1$
0.5000 × 2	1.0000	1 + 0.0000	$a_{-4} = 1$

$(0.6875)_{10} = (.1011)_2$

Example: Decimal to Binary

- Find the binary equivalent of $(23.8125)_{10}$

2	23		
2	11 –	1	LSB
2	5 –	1	
2	2 –	1	
2	1 –	0	
	0 –	1	MSB

$(23)_{10} = (10111)_2$

MSB LSB

	Product	Integer + Fraction	Coefficient
0.8125 × 2	1.6250	1 + 0.6250	$a_{-1} = 1$
0.6250 × 2	1.2500	1 + 0.2500	$a_{-2} = 1$
0.2500 × 2	0.5000	0 + 0.5000	$a_{-3} = 0$
0.5000 × 2	1.0000	1 + 0.0000	$a_{-4} = 1$

$(0.8125)_{10} = (.1101)_2$

$(23.8125)_{10}$

10111.1101