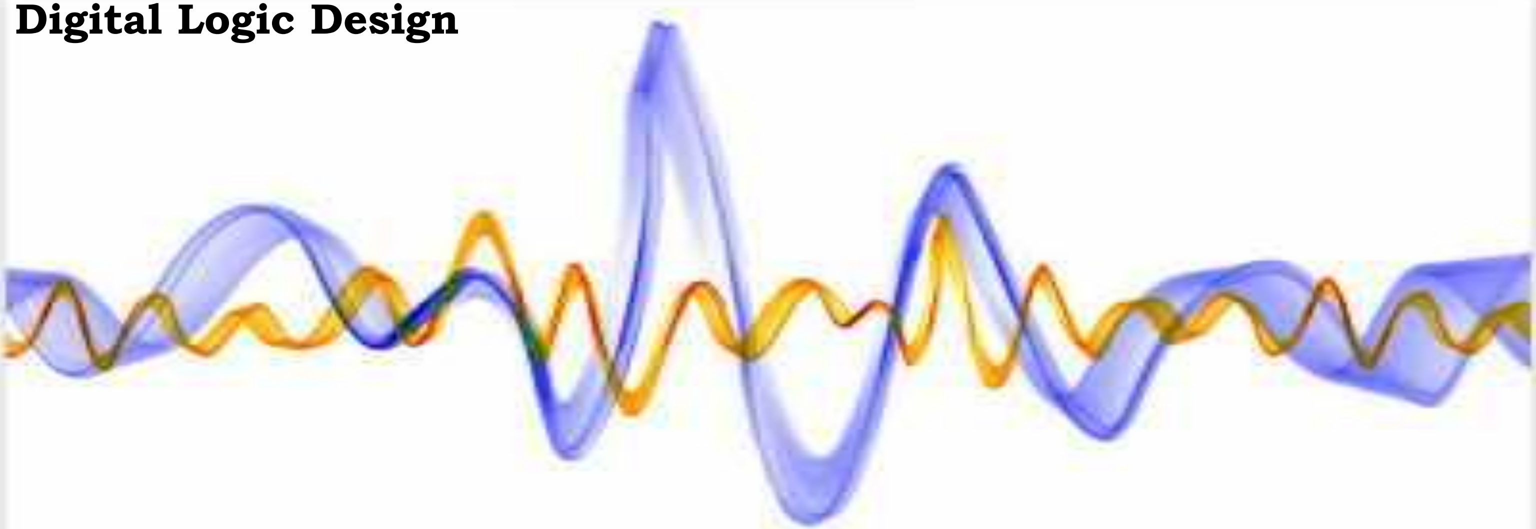
Digital Logic Design



Lecture 1:

Chapter 1: Digital Systems and Binary Numbers

Mirvat Al-Qutt, Ph.D
Computer Systems Department , FCIS,
Ain Shams University

Agenda

- What's this course about?
- Course Arrangement:
  - Study Materials
  - Teaching Methods
  - Lab Activities,
  - Grading and Assessment
  - Syllabus (Planned)
  - Instructor Contact

Logistics

- Lectures
  - Tell & Show you digital logic design concepts
- Tutorial and Lab
  - Exercises and Practical matters
- Assignments
  - Weekly Assignment

What is this course about? What is Logic Design ?

What is design?

- Given a **problem** specification, come up with a systematic way of finding the **solution**, that involves choosing appropriate **components** while meeting some of the design **constraints** such as size, cost, power, beauty, elegance, etc.

What is logic design?

- Determining the collection of **digital logic components and the** interconnections between them to perform a specified **control** and/or **data manipulation** and/or **communication** functions
- The design may need to be optimized and/or transformed to meet design constraints

What is this course about? What is Logic Design ?

Why study Logic Design

- **First step to understand computer architectures** from both hardware and computations perspectives
- It is the base of all modern computing/ control devices
- It makes all the following possible
  - Microprocessors
  - Storage so inexpensive and dense
  - Wireless networking
  - New materials

Study Materials

1. Notes/slides

2. Tutorial / Lab Sheets

3. Textbook
   - Digital Design [5th Edition] (M. Morris Mano and Michael Ciletti) , Download PDF From Here.

Teaching Methods

- Interactive Lecture
- Discussions
- Problem Based learning
- Assignments
- Experimental learning: Lab Activities devoted to practice Digital Design concepts through a series of hands-on

Grading and Assessment

Assessment	Marks
Final Written Exam	**50**
Midterm	**15**
Quizzes	**5**
Lab Activities ,Assignments and Tasks	**10**
Practical Exam	**20**

Syllabus

1. Digital Systems and Binary Numbers
2. Boolean Algebra and Logic Gates
3. Gate – Level Minimization
4. Combinational Logic
5. Synchronous Sequential Logic
6. Registers and Counters

Contact

- Mirvat Al-Qutt, Ph.D.
- Email: mmalqutt@cis.asu.edu.eg

Outlines

- 1.1 Digital Systems
- 1.2 Binary Numbers
- 1.3 Number-base Conversions
- 1.4 Octal and Hexadecimal Numbers
- 1.9 Binary Logic

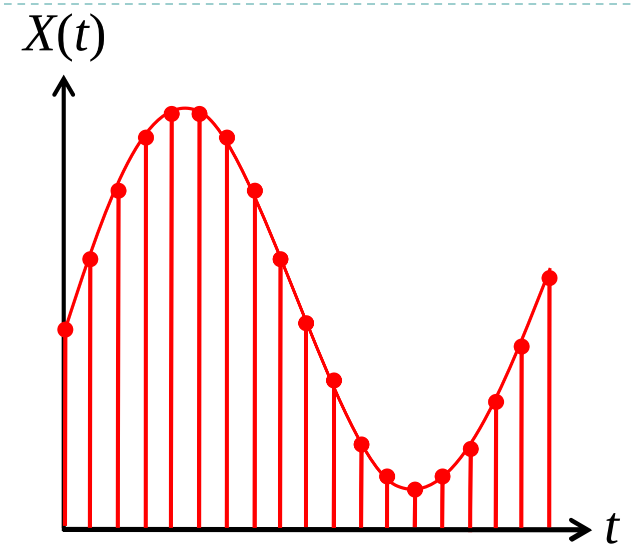
Digital Systems

- One characteristic of digital systems is their ability to represent and manipulate discrete elements of information
  - 10 decimal digits {0,1, 2, 3, …,9}
  - 26 letters of alphabet {A, B, C, …,Z}
  - 64 squares of chessboard

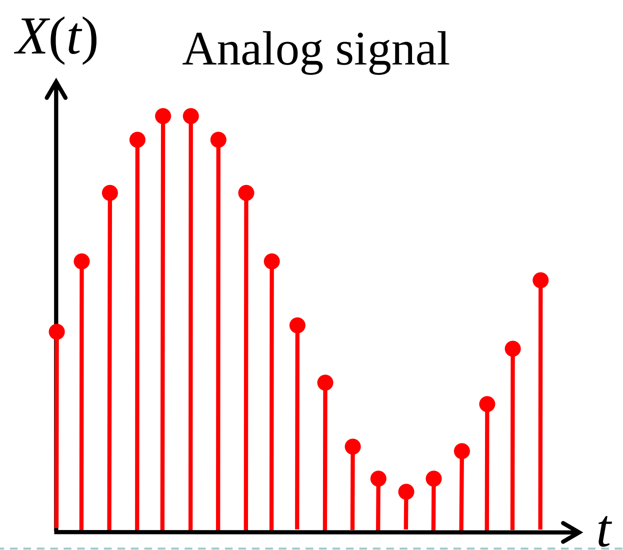
Analog and Digital Signal

- **Discrete quantities of information either emerge from the nature of the data being processed or may be quantized** from a continuous process.

- 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.



Analog signal



Digital signal

Why Digital Systems ?

A World Transformed: What Are the Top 30 Innovations of the Last 30 Years?

Published: February 18, 2009 in Knowledge@Wharton

1. Internet, broadband, WWW (browser and html)
2. PC/laptop computers
3. Mobile phones
4. E-mail
5. DNA testing and sequencing/Human genome mapping
6. Magnetic Resonance Imaging (MRI)
7. Microprocessors
8. Fiber optics
9. Office software (spreadsheets, word processors)
10. Non-invasive laser/robotic surgery (laparoscopy)
11. Open source software and services (e.g., Linux, Wikipedia)
12. Light emitting diodes
13. Liquid crystal display (LCD)
14. GPS systems
15. Online shopping/ecommerce/auctions (e.g., eBay)
16. Media file compression (jpeg, mpeg, mp3)
17. Microfinance
18. Photovoltaic Solar Energy
19. Large scale wind turbines
20. Social networking via the Internet
21. Graphic user interface (GUI)
22. Digital photography/videography
23. RFID and applications (e.g., EZ Pass)
24. Genetically modified plants
25. Bio fuels
26. Bar codes and scanners
27. ATMs
28. Stents
29. SRAM flash memory
30. Anti retroviral treatment for AIDS

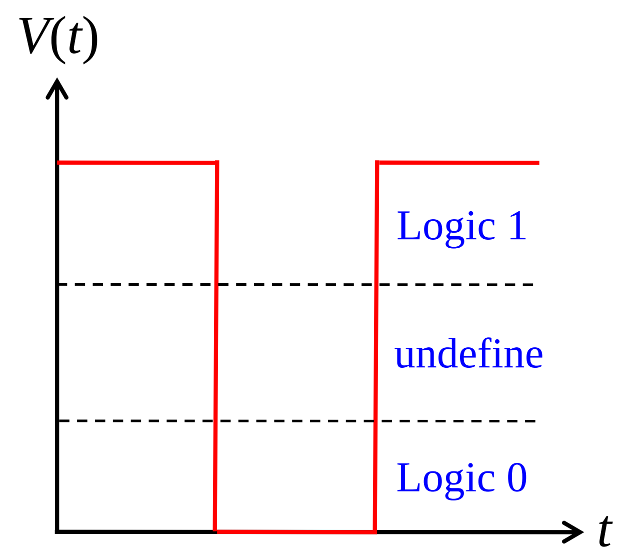
Of these 30 innovations , 10 are directly related to advances in Digital Logic and Solid State Circuits;

Another 8 are the indirect results of ICs.

Binary Digital Signal

- Binary digital systems, the variable takes on discrete values.
  - Two level, or binary values are the most prevalent values.

- Binary values are represented abstractly by:
  - Digits 0 and 1
  - False (F) and True (T)
  - Low (L) and High (H)
  - On and Off



Binary digital signal

Decimal Number System **(base 10)**

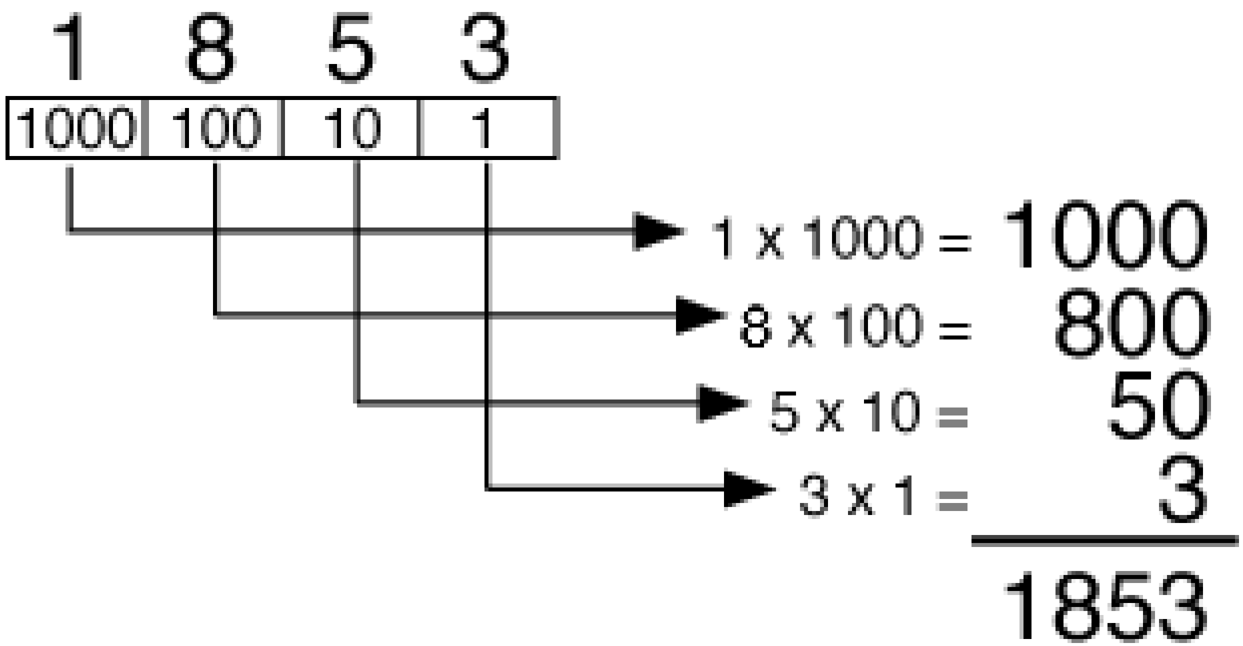
- For solid and deep understanding of binary numbers we recall our understanding of decimal number system with more analysis.
- **Example: 7392**

7	3	9	2	
7	0	0	0	$7*10^3$
0	3	0	0	$+ 3*10^2$
0	0	9	0	$+ 9*10^1$
0	0	0	2	$+ 3*10^0$

The power of 10 is implied by the digit (coefficient) position

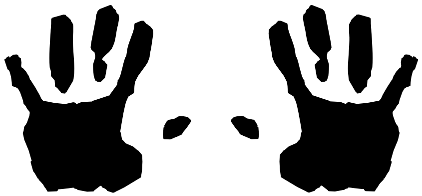
Decimal Number System

- For solid and deep understanding of binary numbers we recall our understanding of decimal number system with more analysis.
- **Example: 1853**



Decimal Number System

- **Base (also called radix) = 10**
  - 10 digits { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }
  - 10 possible digits ranges from (0 to r-1)



5	1	2	.	5	4

- **Digit Position**
  - Integer & fraction

10^2	10^1	10^0	.	10^{-1}	10^{-2}
100	10	1	.	0.1	0.01

Weights

- **Digit Weight**
  - Weight = $(Base=10)^{Position}$

- **Magnitude**
  - Sum of "*Digit Value* x *Weight*"

500	10	2	.	0.5	0.04

$d_2*B^2+d_1*B^1+d_0*B^0+d_{-1}*B^{-1}+d_{-2}*B^{-2}$

(512.54)10

- **Formal Notation** $(\ldots)_{10}$

Binary Number System **(Base 2)**

- **Base (also called radix) = 2**
  - **2 digits { 0, 1)**
  - **2 possible digits ranges from (0 to r-1)**

1	1	0	1	0	.	1	1

- **Digit Position**
  - **Integer & fraction**

2^4	2^3	2^2	2^1	2^0	.	2^{-1}	2^{-2}
16	8	4	2	1	.	0.5	0.25

Weights

- **Digit Weight**
  - **Weight = $(Base=2)^{Position}$**

- **Magnitude (Decimal Equivalent)**
  - **Sum of "*Digit* x *Weight*"**

16*1 + 8*1+4*0+ 2*1+1*0+1*0.5+1*0.25

$=(26.75)_{10}$

- **Formal Notation $(\ldots)_2$**

Base - 5 Number System

- **Base (also called radix) = 5**
  - **5 digits { 0, 1,2,3,4)**
  - **5 possible digits ranges from (0 to r-1)**
- **Digit Position**
  - **Integer & fraction**

4	0	2	1	.	2

- **Digit Weight**
  - **Weight = (*Base=5*)** ^{Position}

5^3	5^2	5^1	5^0	.	5^{-1}
125	25	5	1	.	0.2

Weights

- **Magnitude (Decimal Equivalent)**
  - **Sum of "*Digit* x *Weight*"**

125*4 + 25*0+5*2+1*1+2*0.2

$=(511.4)_{10}$

- **Formal Notation** $(\ldots)_5$

Base – 8(**Octal**) Number System

- **Base (also called radix) = 8**
  - **8 digits { 0, 1,2,3,4,5,6,7)**
  - **8 possible digits ranges from (0 to r-1)**

1	2	7	.	4

- **Digit Position**
  - **Integer & fraction**

8^2	8^1	8^0	.	8^{-1}
64	8	1	.	0.125

Weights

- **Digit Weight**
  - **Weight = (*Base*)** ***Position***
- **Magnitude (Decimal Equivalent)**
  - **Sum of "*Digit* x *Weight*"**

64*1 + 8*2+1*7+0.125*4

$=(87.5)_{10}$

- **Formal Notation** $(\ldots)_8$

Base – 16 (**Hexadecimal**) Number System

- **Base (also called radix) = 16**
  - **16 digits { 0, 1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)**
  - **16 possible digits ranges from (0 to r-1)**
  - **The letters of the alphabet are used to supplement the 10 decimal digits when the base of the number is greater than 10.**
- **Digit Position**
  - **Integer & fraction**
- **Digit Weight**
  - **Weight = (*Base=16*) *Position***
- **Magnitude (Decimal Equivalent)**
  - **Sum of "*Digit x Weight*"**
- **Formal Notation $(\ldots)_{16}$**

B	6	5	F

16^3	16^2	16^1	16^0

$$16^3 * B + 16^2 * 6 + 16^1 * 5 + 16^0 * F$$

$$16^3 * (11) + 16^2 * 6 + 16^1 * 5 + 16^0 * (15)$$

$$= (46{,}687)_{10}$$

Hexadecimal System

- The hexadecimal system is used commonly by designers to represent long strings of bits in the addresses, instructions, and data in digital systems.
- For example

$1*2^3$	$0*2^2$	$1*2^1$	$1*2^0$
8+0+2+1 =11 = B			

$1*2^3$	$1*2^2$	$1*2^1$	$1*2^0$
8+4+2+1 =15 = F			

1011	0110	0101	1111

B	6	5	F

More about Binary System

- The digits in a binary number are called **bits**.
- **When a bit is equal to 0**, it does not contribute to the sum during the conversion.
- Therefore, the **conversion** from **binary to decimal** can be obtained by adding only the numbers with powers of two corresponding to the bits that are equal to 1

1	1	0	1	0

2^4	2^3	2^2	2^1	2^0
16	8	4	2	1

$16 + 8 + 2 = (26)_{10}$

More about Binary System

- The **conversion** from **binary to decimal**
  1. Write binary number
  2. Write place heading
  3. Ignore zeros
  4. Sum up headings mapped to 1's only

Place headings

128	64	32	16	8	4	2	1
0	1	0	1	1	1	0	1

Binary number

=> 64 + 16 + 8 + 4 + 1 = 93

More about Binary System (units)

- In computer work,
- $\mathbf{2^{10}}$ is referred to as K (**kilo**), <u>4K = 2^{12} = 4,096</u>
- $\mathbf{2^{20}}$ as M (**mega**), and <u>16M = 2^{24} = 16,777,216</u>
- $\mathbf{2^{30}}$ as G (**giga**), 4G = 2^{32} bytes
- $\mathbf{2^{40}}$ as T (**tera**).
- Computer capacity is usually given in bytes. A *byte* is equal to eight bits and can accommodate

More about Binary System (units)

- Computer capacity is usually given in bytes. A *byte* is equal to eight bits and can accommodate

Unit	Bytes
1 Bit	0,1
1 Byte	8 bits
1 Kilobyte (Kb)	2^{10} = 1024 bytes
1 Megabyte (Mb)	2^{20} = 1,048,576 bytes (1024 Kb)
1 Gigabyte (Gb)	2^{30} = 1,073,741,824 bytes (1024 Mb)
1 Terabyte (Tb)	2^{40} = 1,099,511,627,776 bytes (1024 Gb)

More about Binary System (Range)

- These measurements are used to determine the **lower and upper limits** of the range numbers possible with a given amount of bits (vise versa)

Place headings

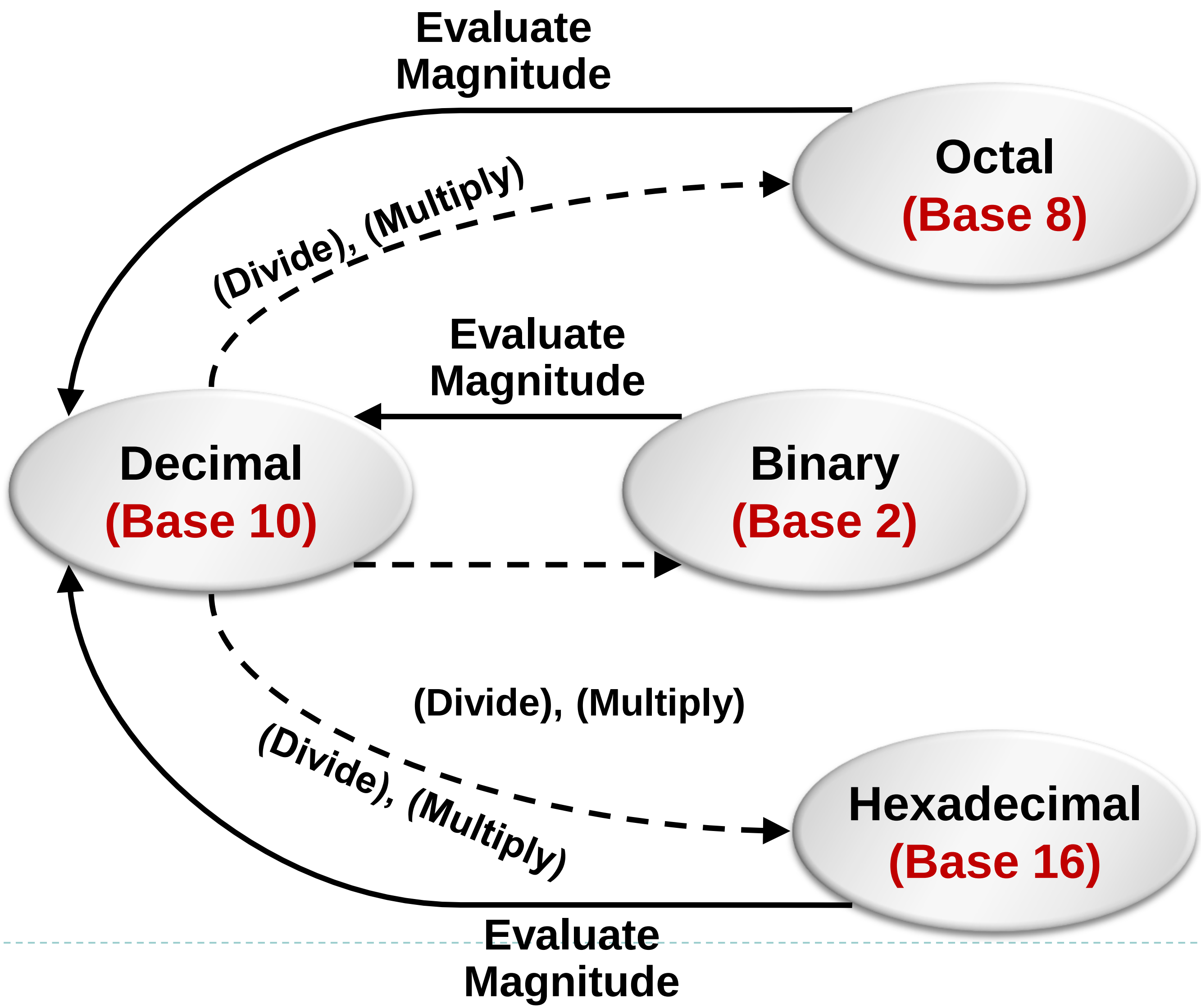
128	64	32	16	8	4	2	1
0	1	0	1	1	1	0	1

Binary number

=> 64 + 16 + 8 + 4 + 1 = 93

Unit	Range
1 Bit	0 to $2^1 - 1$ (0 to 1)
8 bits (1 Byte)	0 to $2^8 - 1$ (0 to 255)
16 bits (2 bytes)	0 to $2^{16} - 1$ (0 to 65,535)
24 bits (3 bytes)	0 to $2^{24} - 1$ (0 to 16, 777, 215)
32 bits (4 bytes)	0 to $2^{32} - 1$ (0 to 4,294,967,295)

Number Base Conversions



Decimal (***Integer***) to Binary Conversion

- **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
13/ 2 =	6	1
6 / 2 =	3	0
3 / 2 =	1	1
1 / 2 =	0	1

Answer: $(13)_{10} = (1101)_2$

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: $(175)_{10}$

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

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 Hexadecimal Conversion

Example: $(175)_{10}$

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

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

Example: $(0.3125)_{10}$

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

Answer: $(0.3125)_{10} = (0.a_{-1})_{16} = (0.5)_{16}$

Try it yourself

- Convert 41 decimal to binary

The arithmetic process can be manipulated more conveniently as follows:

Integer	Remainder	
41		
20	1	
10	0	
5	0	
2	1	
1	0	
0	1	101001 = answer

Try it yourself

- Convert 153 decimal to octal

153	
19	1
2	3
0	$2 = (231)_8$

Try it yourself

- Convert 0.6875 decimal to binary

	Integer		Fraction	Coefficient
$0.6875 \times 2 =$	1	+	0.3750	$a_{-1} = 1$
$0.3750 \times 2 =$	0	+	0.7500	$a_{-2} = 0$
$0.7500 \times 2 =$	1	+	0.5000	$a_{-3} = 1$
$0.5000 \times 2 =$	1	+	0.0000	$a_{-4} = 1$

Therefore, the answer is $(0.6875)_{10} = (0.\, a_{-1}\, a_{-2}\, a_{-3}\, a_{-4})_2 = (0.1011)_2$.

Try it yourself

- Convert 0.513 to octal

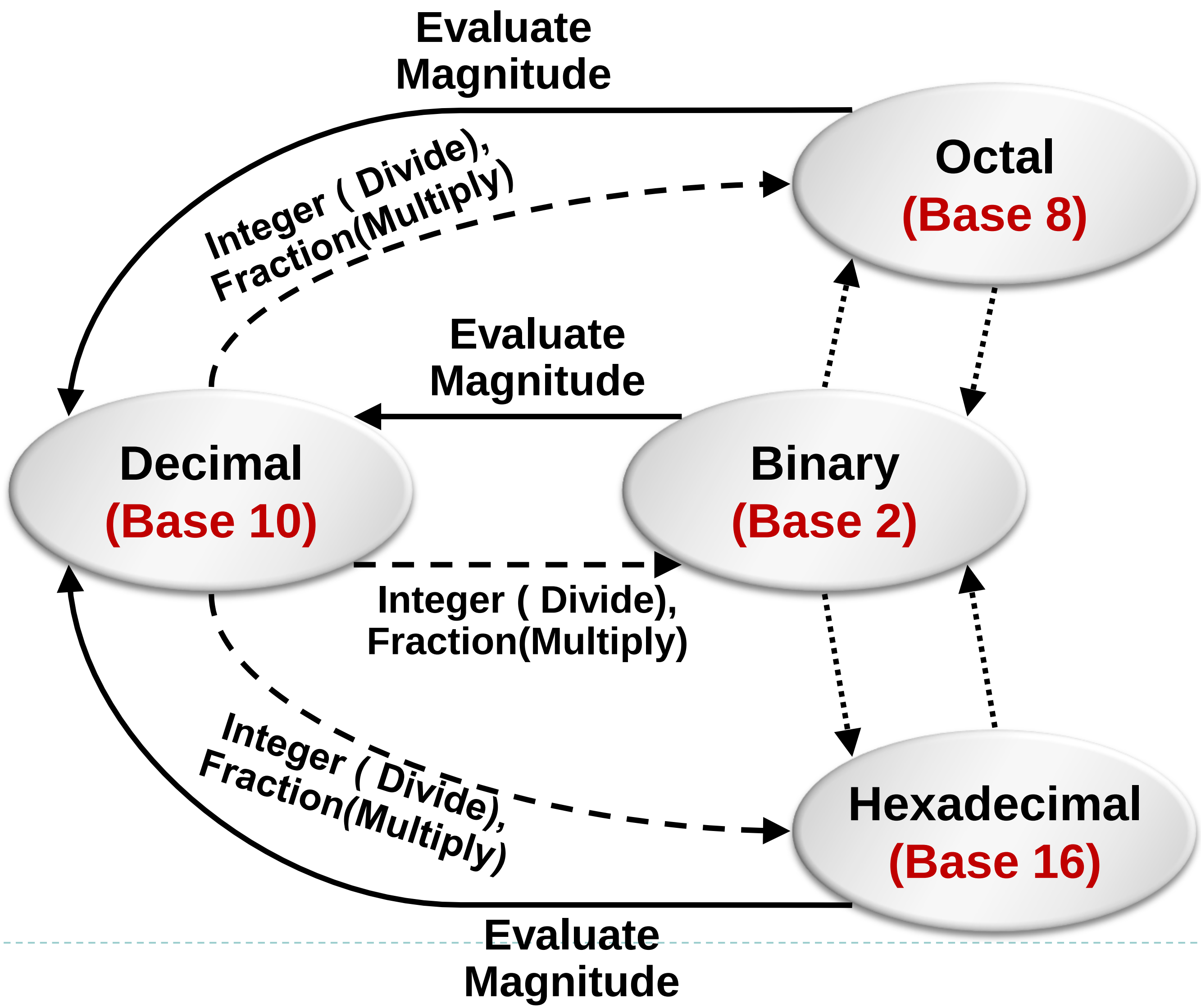
Convert $(0.513)_{10}$ to octal.

$$0.513 \times 8 = 4.104$$
$$0.104 \times 8 = 0.832$$
$$0.832 \times 8 = 6.656$$
$$0.656 \times 8 = 5.248$$
$$0.248 \times 8 = 1.984$$
$$0.984 \times 8 = 7.872$$

The answer, to seven significant figures, is obtained from the integer part of the products:

$$(0.513)_{10} = (0.406517\ldots)_8$$

Number Base Conversions



Binary – Octal Conversion

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

Example:

$$(10110.01)_2$$

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

Binary – Octal Conversion

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

Example:

Assume Zeros

$$(\underbrace{0\,1\,0}\,\underbrace{1\,1\,0}\,.\,\underbrace{0\,1\,0})_2$$

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

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 – Hexadecimal Conversion

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

Example:

Assume Zeros

$(0001\ 0\ 1\ 1\ 0\ .\ 0\ 1\ 00)_2$

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

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

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

Octal – Hexadecimal Conversion

- 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)

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

Try it yourself

- Convert (01101011. 111100) **binary** to **octal**

01	101	011	.	111	100

1	5	3	.	7	4

- Convert (01101011. 111100) **binary** to **Hexadecimal**

0110	1011	.	1111	00

6	B	.	F	0

Try it yourself

- Convert (673.12) **octal** to **binary**

6	7	3	.	1	2
110	111	011	.	001	010

- Convert (306.D) **Hexadecimal** to **binary**

3	0	6	.	D
0011	0000	0110	.	1101


Thank You!