



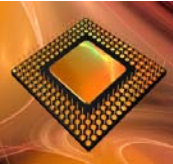
**WELCOME TO**

# **Modern Digital System Design**

**ECE 2372 / Fall 2018 / Lecture 01**

**Texas Tech University  
Dr. Tooraj Nikoubin**

**Introduction,  
Number Systems and Conversion**



# Grading Scheme



Course Requirements and Corresponding Weight		
1	Test # 1	15%
2	Test # 2	15%
3	Final exam	30%
4	Project	20%
5	Homework and Quiz	20%

Bonus ?%

# Grading and Scheme

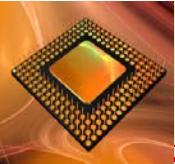


# Class attending policy

---



1. Efficient study and class attention
2. Cellphone & Laptop ?
3. Class activities (Quiz, Present & absence )
4. Polite or shy ?
5. Challenge or stress ?
6. Who is Brave ?
7. Sample of Tests

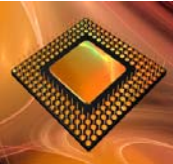


# Main Sources

---



- 1-M.M. Mano and C.R. Kime,  
"Logic and Computer Design Fundamentals" 4 th Edition,  
Pearson -Prentice Hall.
- 2-Charles H.Roth, Jr. and Larry L. Kinney,  
**" Fundamentals of Logic Design"**

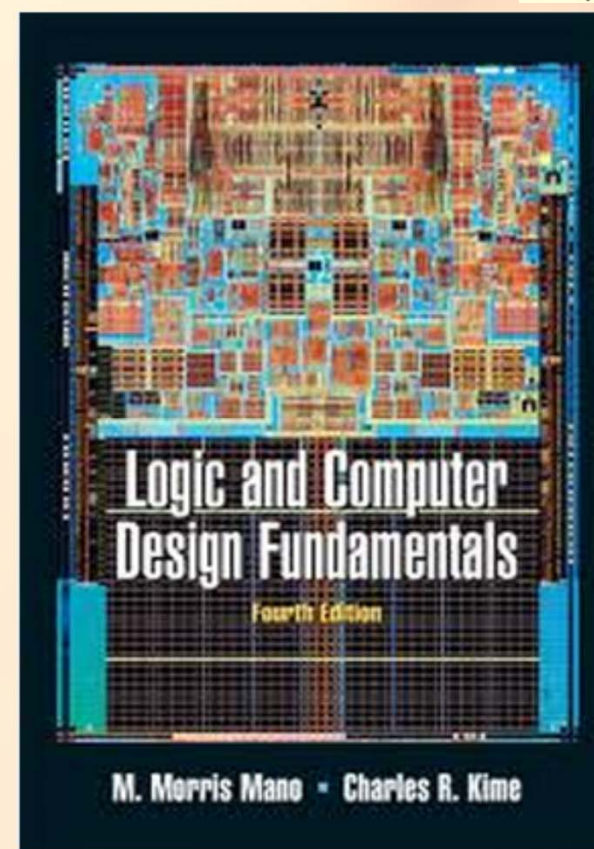
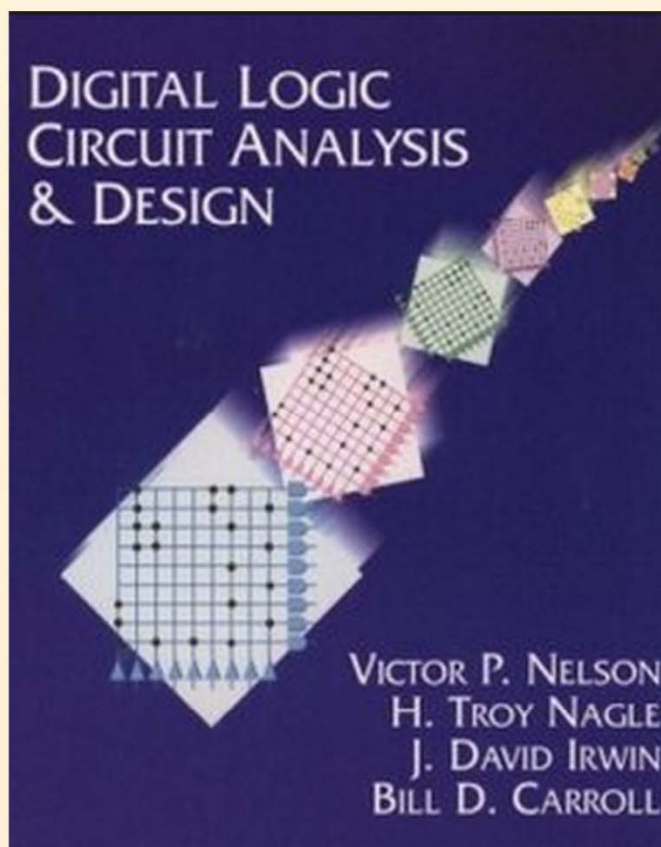
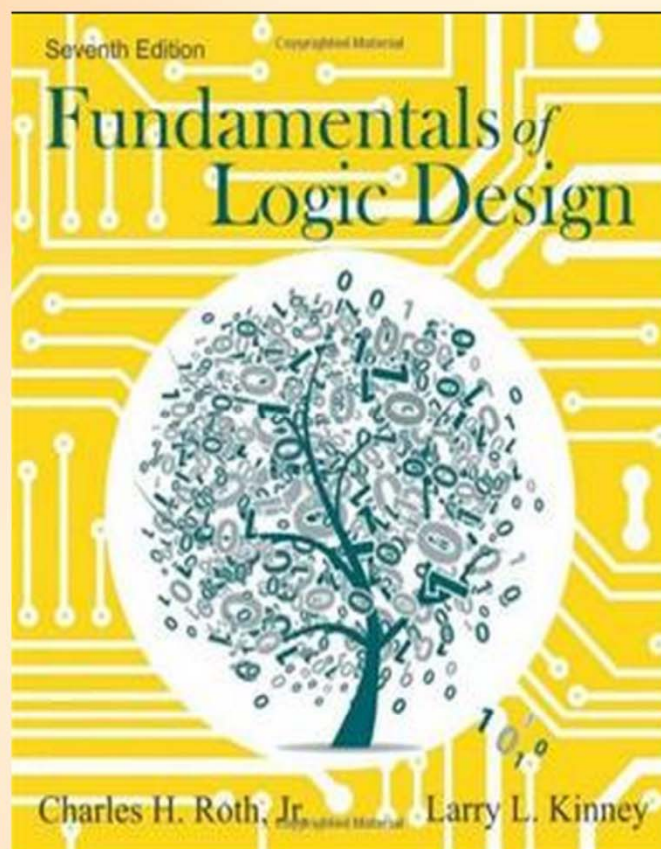
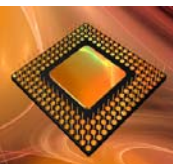


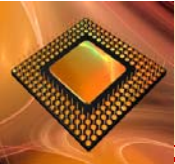
## Other References



1. Victor P. Nelson, H. Troy Nagle, Bill D. Carroll, David Irwin  
**“Digital Logic Circuit Analysis and Design ”**
2. M. Mano, **“*Digital Design*”**, 3rd Edition, Prentice Hall, Upper-Saddle River, New Jersey, 2002
3. Nazeib M. Botros,  
**“HDL programming Fundamentals VHDL and Verilog”**
4. Stephen Brown and Zvonko Vranesic,  
**“*Fundamentals of Digital Logic with Verilog Design*”**  
, McGraw-Hill, 2003







# Main Sources for the test

---



**1-PowerPoint slides**

**2-Charles H.Roth, Jr. and Larry L. Kinney, " Fundamentals of Logic Design"**

**3-M.M. Mano and C.R. Kime, "Logic and Computer Design Fundamentals" 4 th Edition, Pearson -Prentice Hall.**

-----

**4. Homework**

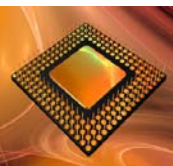
-----

**5. Quiz**

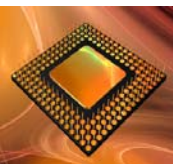
-----

**6. Sample of test**





Topic	Lectures
• Introduction, Number Systems and Conversion	1L
• Logic Gates and Boolean Algebra	1L
• Applications of Boolean Algebra, Minterm and Maxterm Expansions	2L
• Multi- Level Gate Circuits , NAND and NOR Gates	1L
• Karnaugh Maps and Quine-McClusky Method for simplification	1L
<< Review and Test #1 >>	1L
• Combinational Circuit Design,	1L
• Multiplexers, Decoders, ROM and Programmable logic Devices	2L
• Add and Sub, Adders, Subtractors and Comparators	1L
• Coding	1L
• Hardware description language for combinational circuits	1L
<< Review and Test #2 >>	1L
• Latch and Flip-Flops	2L
• Registers and Buffers	1L
• Counters and Counter circuit design	2L
• Analysis of clocked Sequential Circuits	2L
• Sequential Circuit Design	2L
• State machine design	2L
• Arithmetic Circuits	1L
• Hardware description language for sequential circuits	1L
<< Review and Final exam >>	1L
L=75min	Total 28L



# ECE 2372 ( Modern Digital System Design )

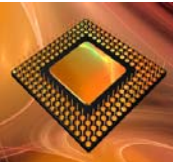


TA: TBA  
Tutors: TBA

Email:  
Office Hours:

	8	9	10	11	12	1	2	3
Monday								
Tuesday								
Wednesday								
Thursday								
Friday								
Saturday								
Sunday								

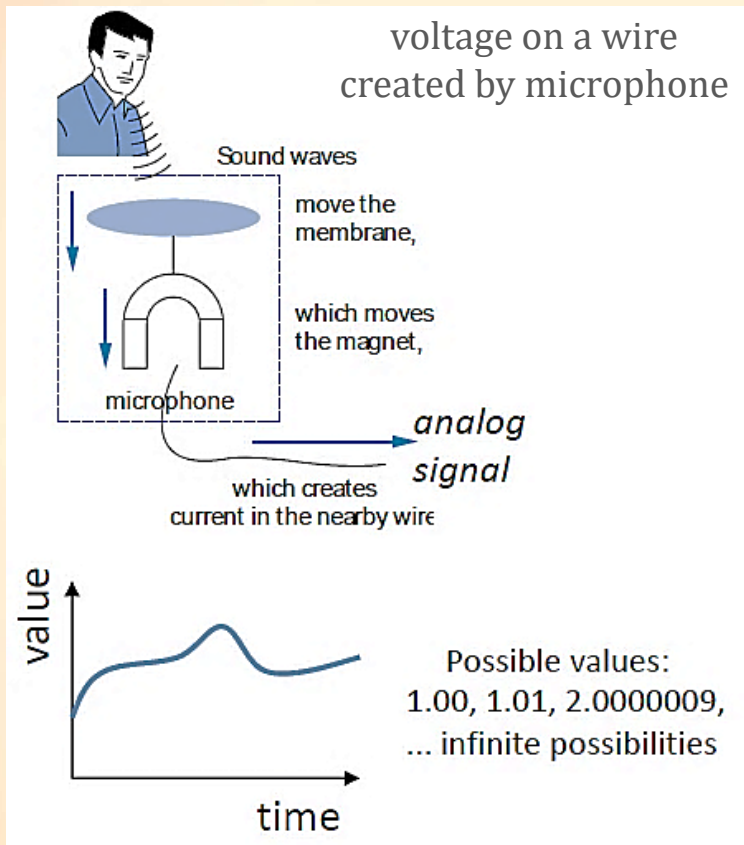
Office : ECE Computer Lab



# Digital and Analog Signals

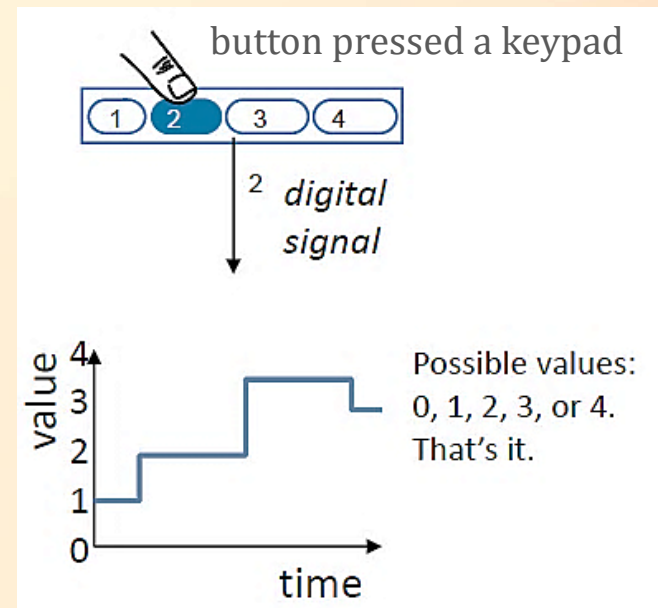
## Analog signal

With Infinite possible values

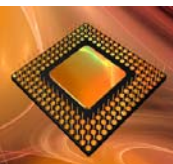


## Digital signal

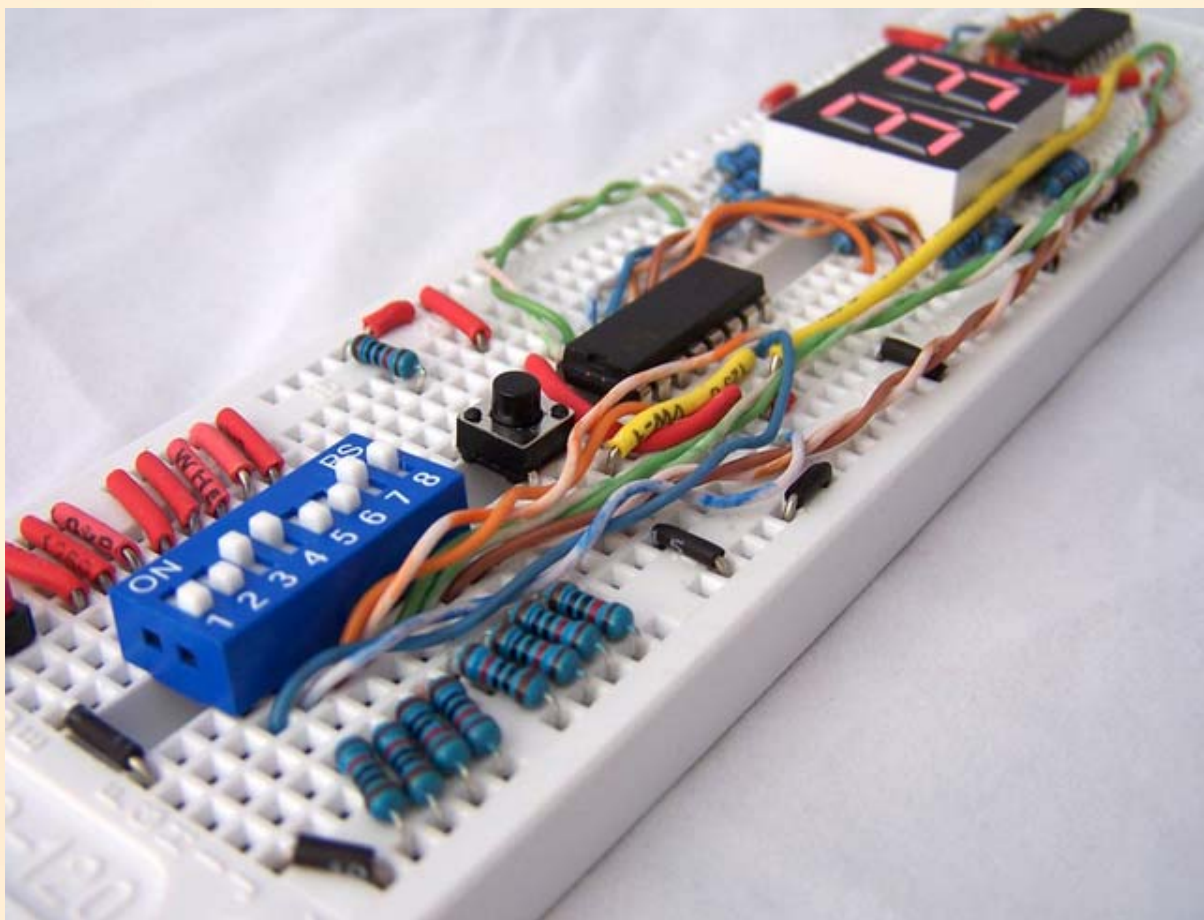
With Finite possible values



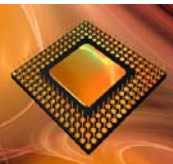




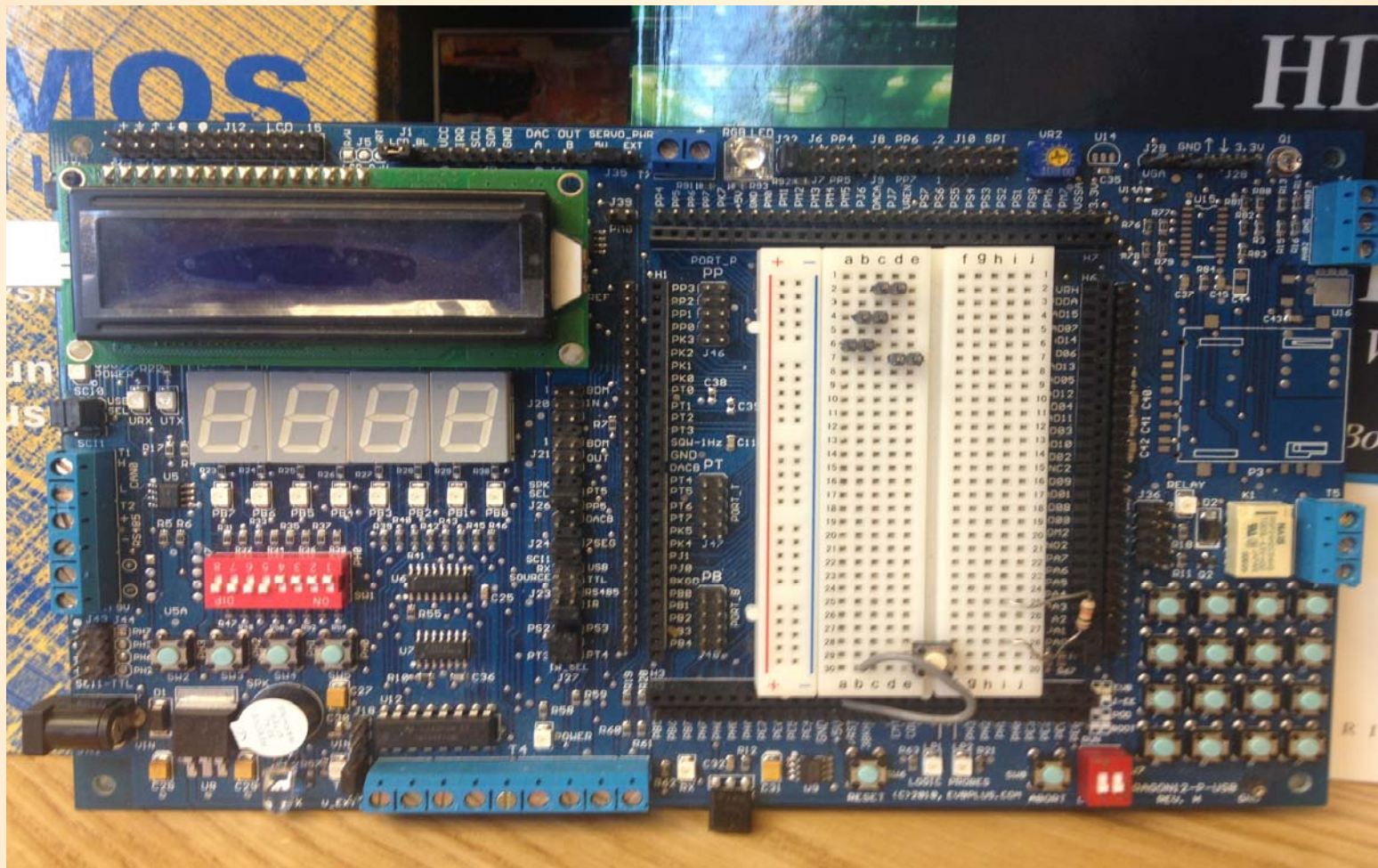
# A Sample of Digital Board



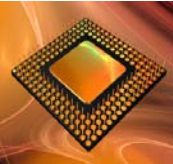




# A Sample of Digital Board







# Digital Design



- **What is digital ?**

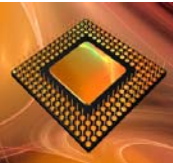
- Digital camera, Digital TV, Digital Watch, Digital Radio, Digital City (e-city), Digital Photo Frame ...etc
- Which gives the things in countable form
- Scene (analog) to Image (digital)

- **Why digital ?**

- Countable form, makes easy to manage
- Easy management makes more useful and versatile

- **What digit ?**

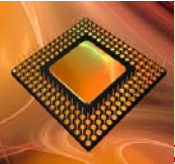
- How to count: Decimal digit: 0 to 9



# What Digit? => Number System



- Famous Number System: Dec, Rom, Bin
- Decimal System: 0 -9
  - May evolves: because human have 10 finger
- Roman System
  - May evolves to make easy to look and feel
  - Pre/Post Concept: (IV, V & VI) is (5-1, 5 & 5+1)
- Binary System, Others (Oct, Hex)
  - One can cut an apple in to two



# Design & Logic Design



- **What is design?**

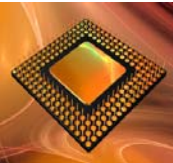
- Given problem spec, solve it with available components
- While meeting quantitative (size, cost, power) and qualitative (beauty, elegance)

- **What is logic design?**

- Choose digital logic components to perform specified control, data manipulation, or communication function and their interconnection
- Which logic components to choose?

Many implementation technologies (fixed-function components, *programmable devices*, individual transistors on a chip, etc.)

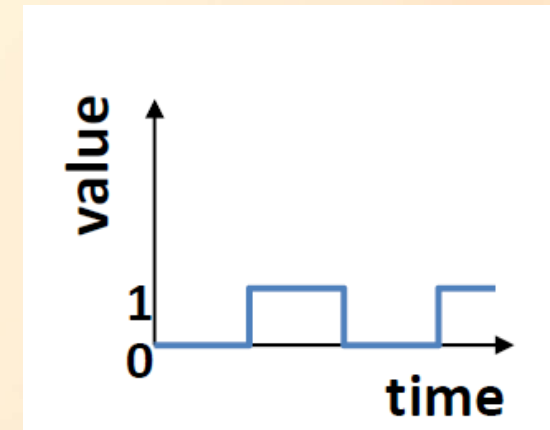
- Design optimized/transformed to meet design constraints



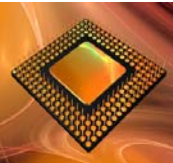
# Digital Signals with Only Two Values: Binary



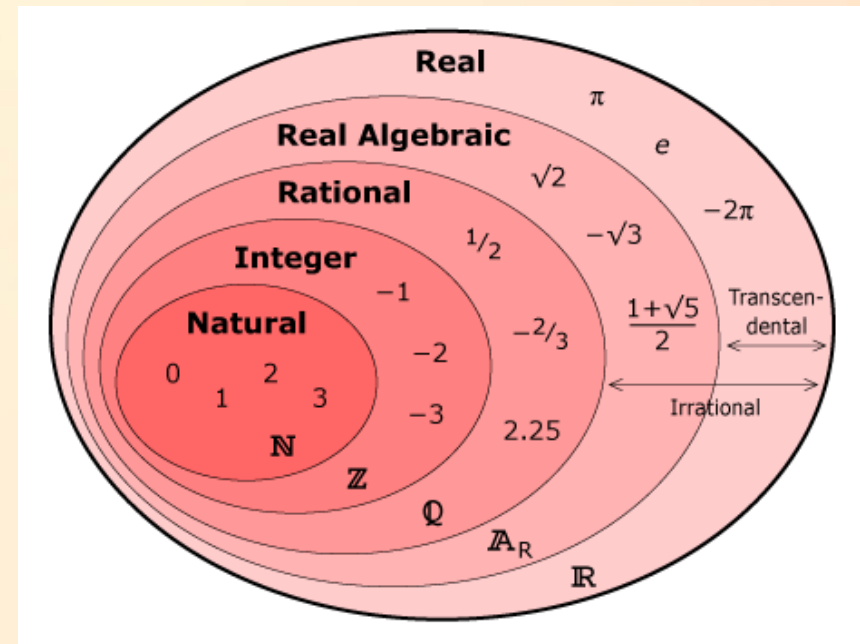
- *Binary* digital signal -- only *two* possible values
  - Typically represented as **0** and **1**
- One **B**inary **dig**it is **BIT** value
  - We'll only consider *binary* digital signals
  - Binary is popular because

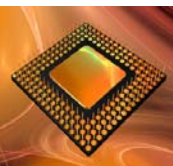


- Transistors, the basic digital electric component, operate at *two* states (switch on and switch off )
- Storing/transmitting one of *two* values is easier than three or more (e.g., loud beep or quiet beep, reflection or no reflection)



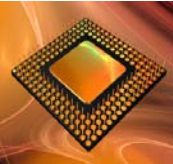
# Number Systems and Conversion





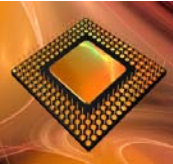
Decimal	Binary	Octal	Hexadecimal
0	00000	0	0
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10



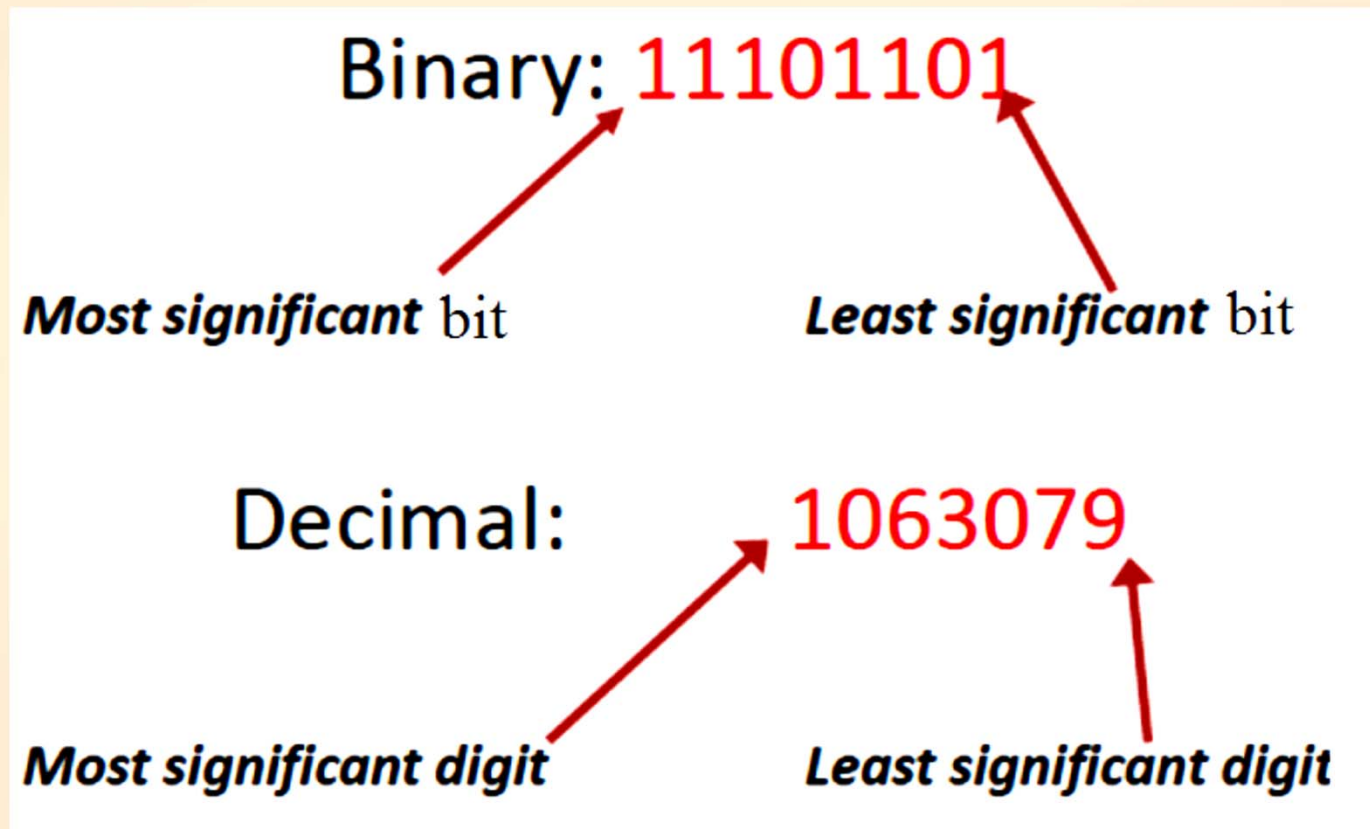


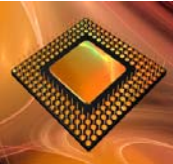
# Outline

- Number System  
Decimal, Binary, Octal, Hex
- Conversion (one to another)  
Decimal to Binary, Octal, Hex & Vice Versa  
Binary to HEX & vice versa
- Other representation  
Signed, Unsigned, Complement



# Significant Digits

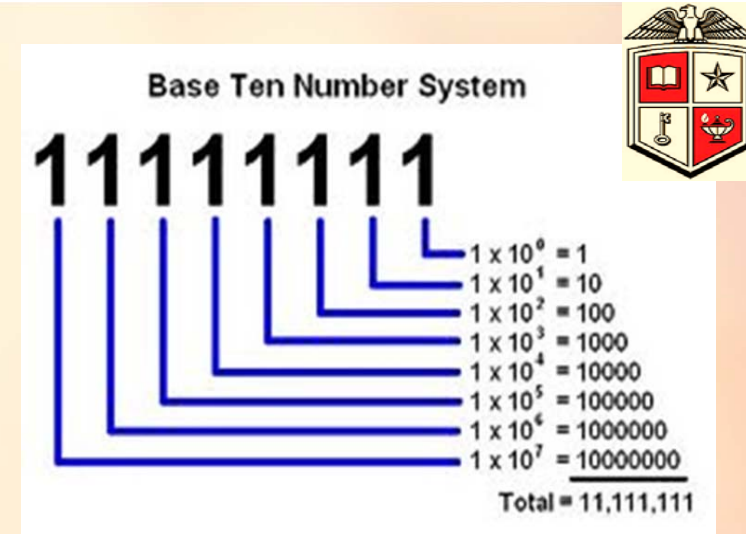




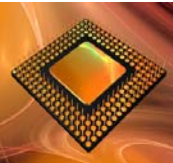
# Decimal (base 10)

- Uses positional representation
- Each digit corresponds to a power of 10 based on its position in the number
- The powers of 10 increment from 0, 1, 2, etc. as you move right to left

$$1,586 = 1 * 10^3 + 5 * 10^2 + 8 * 10^1 + 6 * 10^0$$







# Binary to Decimal

0	0	1	1	1		
			1	x 1	=	1
			1	x 2	=	2
		1	x 4	=	4	
	0	x 8	=	0		
0	x 16	=	0			

**Answer:** 00111 = 7

1	0	1	0	0		
			0	x 1	=	0
			0	x 2	=	0
		1	x 4	=	4	
	0	x 8	=	0		
1	x 16	=	16			

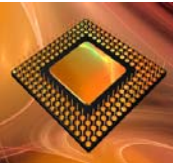
**Answer:** 10100 = 20

0	1	0	1	1		
			1	x 1	=	1
			1	x 2	=	2
		0	x 4	=	0	
	1	x 8	=	8		
0	x 16	=	0			

**Answer:** 01011 = 11

1	1	0	1	1		
			1	x 1	=	1
			1	x 2	=	2
		0	x 4	=	0	
	1	x 8	=	8		
1	x 16	=	16			

**Answer:** 11011 = 27



# How to Encode Numbers: Binary Numbers



## Working with binary numbers

In base ten, helps to know powers of 10

one, ten, hundred, thousand, ten thousand, ...

In base two, helps to know powers of 2

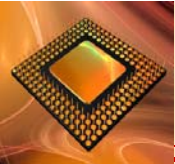
one, two, four, eight, sixteen, thirty two, sixty four, one hundred twenty eight

- Count up by powers of two



$2^9$	$2^8$	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
—	—	—	—	—	—	—	—	—	—
512	256	128	64	32	16	8	4	2	1



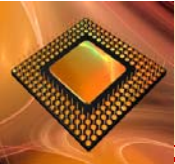


# Octal (base 8)



- Shorter & easier to read than binary
- 8 digits: 0, 1, 2, 3, 4, 5, 6, 7,
- Octal numbers

$$\begin{aligned} 136_8 &= 1 * 8^2 + 3 * 8^1 + 6 * 8^0 \\ &= 1 * 64 + 3 * 8 + 6 * 1 \\ &= 94_{10} \end{aligned}$$

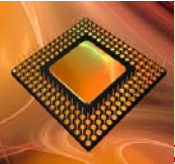


# Hexadecimal (base 16)



- Shorter & easier to read than binary
- 16 digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F
- “0x” often precedes hexadecimal numbers

$$\begin{aligned} 0x123 &= 1 * 16^2 + 2 * 16^1 + 3 * 16^0 \\ &= 1 * 256 + 2 * 16 + 3 * 1 \\ &= 256 + 32 + 3 \\ &= 291 \end{aligned}$$



# Fractional Number

Point: Decimal Point, Binary Point, Hexadecimal point

Decimal

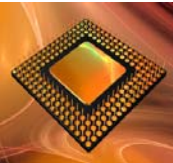
$$247.75 = 2 \times 10^2 + 4 \times 10^1 + 7 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2}$$

Binary

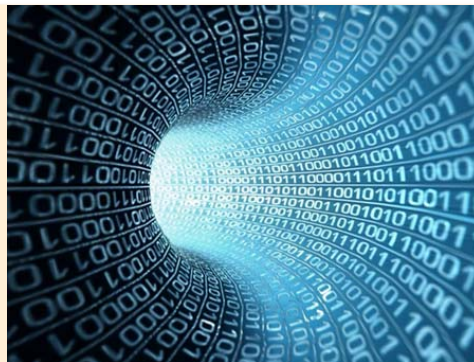
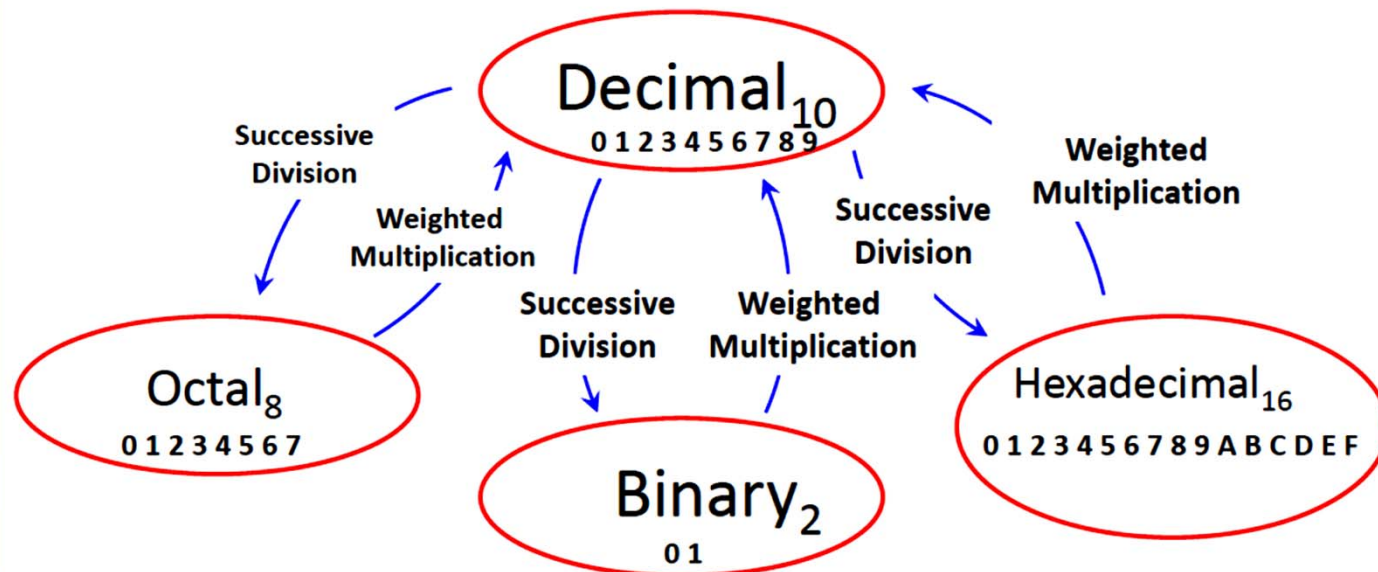
$$10.101 = 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

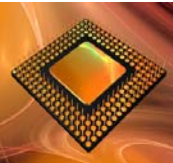
Hexadecimal

$$6A.7D = 6 \times 16^1 + 10 \times 16^0 + 7 \times 16^{-1} + D \times 16^{-2}$$



# Converting To and From Decimal





# Decimal $\leftrightarrow$ Binary

Decimal (Base 10)

Successive  
Division

Binary (Base 2)

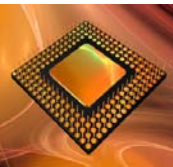
- a) Divide the decimal number by **2**; the remainder is the LSB of the **binary** number.
- b) If the quotient is zero, the conversion is complete. Otherwise repeat step (a) using the quotient as the decimal number. The new remainder is the next most significant bit of the **binary** number.

Binary (Base 2)

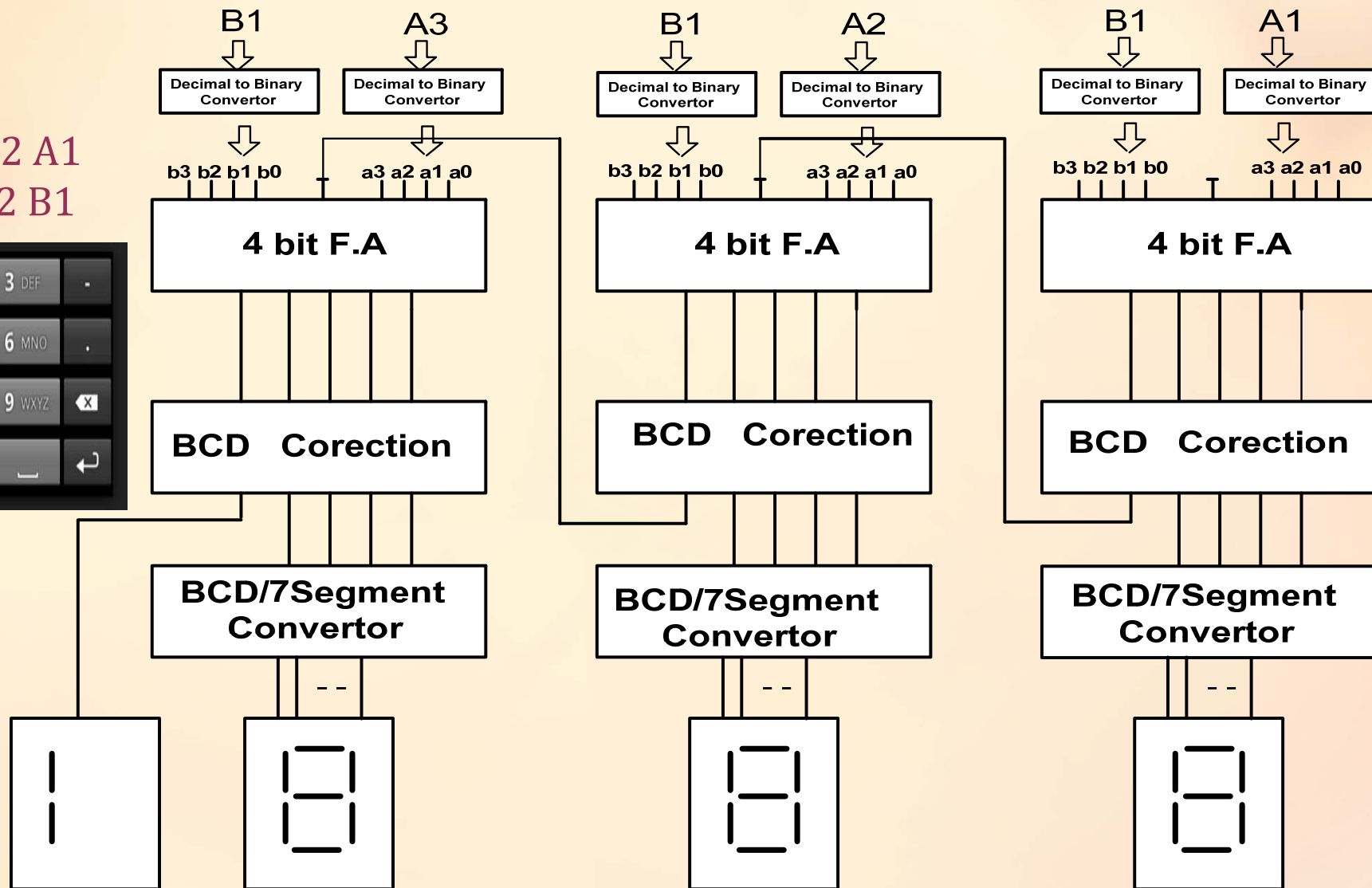
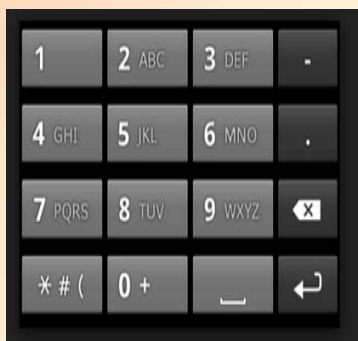
Weighted  
Multiplication

Decimal (Base 10)

- a) Multiply each bit of the **binary** number by its corresponding bit- **Multiplication** weighting factor (i.e., Bit-0  $\rightarrow 2^0=1$ ; Bit-1  $\rightarrow 2^1=2$ ; Bit-2  $\rightarrow 2^2=4$ ; etc).
- b) Sum up all of the products in step (a) to get the decimal number.



A3 A2 A1  
+ B3 B2 B1





# Decimal to Binary : Subtraction Method

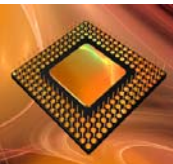


- Goal

- Good for human
- Get the binary weights to add up to the decimal quantity
  - Work from left to right
  - (Right to left – may fill in 1s that shouldn't have been there – try it).

Desired decimal number: **12**

<u>  </u>	<u>  </u>	<u>  </u>	<u>  </u>	<u>  </u>	<u>  </u>	
32	16	8	4	2	1	
<u>1</u>	<u>  </u>	<u>  </u>	<u>  </u>	<u>  </u>	<u>  </u>	=32
32	16	8	4	2	1	too much
<u>0</u>	<u>1</u>	<u>  </u>	<u>  </u>	<u>  </u>	<u>  </u>	=16
32	16	8	4	2	1	too much
<u>0</u>	<u>0</u>	<u>1</u>	<u>  </u>	<u>  </u>	<u>  </u>	=8
32	16	8	4	2	1	ok, keep going
<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>  </u>	<u>  </u>	=8+4=12
32	16	8	4	2	1	DONE
<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>0</u>	
32	16	8	4	2	1	answer

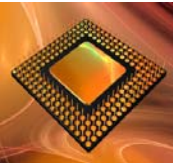


# Decimal to Binary : Subtraction Method



Examples: 39, 27, 18, 7

32	16	8	4	2	1
32	16	8	4	2	1
32	16	8	4	2	1
32	16	8	4	2	1
32	16	8	4	2	1
32	16	8	4	2	1



# Decimal to Binary : Division Method



- Good for computer: Divide decimal number by 2 and insert remainder into new binary number.
- Continue dividing quotient by 2 until the quotient is 0.
- Example: Convert decimal number 12 to binary

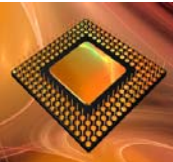
$$12 \text{ div } 2 = ( \text{Quo}=6 , \text{Rem}=0 ) \text{ LSB}$$

$$6 \text{ div } 2 = ( \text{Quo}=3 , \text{Rem}=0 )$$

$$3 \text{ div } 2 = ( \text{Quo}=1 , \text{Rem}=1 )$$

$$1 \text{ div } 2 = ( \text{Quo}=0 , \text{Rem}=1 ) \text{ MSB}$$

$$12_{10} = 1100_2$$



# Conversions Process Decimal $\leftrightarrow$ Base (n)



Decimal (Base 10)

Successive  
Division

Any Base (Base n)

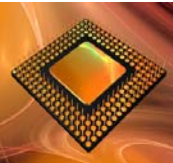
- a) Divide the decimal number by **n**; the remainder is the LSB of the **any base** number.
- b) If the quotation is zero, the conversion is complete. Otherwise repeat step (a) using the quotation as the decimal number. The new remainder is the next most significant bit of the **any base** number.

Any Base (Base n)

Weighted  
Multiplication

Decimal (Base 10)

- a) Multiply each bit of the **any base** number by its corresponding bit- **Multiplication** weighting factor (i.e., Bit-0  $\rightarrow n^0=1$ ; Bit-1  $\rightarrow n^1=n$ ; Bit-2  $\rightarrow n^2=4$ ; etc).
- b) Sum up all of the products in step (a) to get the decimal number.



# Decimal $\leftrightarrow$ Octal Conversion



The Process: Successive Division

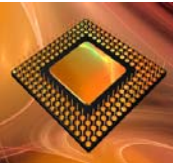
- Divide number by **8**; R is the LSB of the **octal** number
- While Q is 0
- Using the Q as the decimal number.
- New remainder is MSB of the **octal** number.

$$8 \overline{) 94} \quad r = 6 \leftarrow \text{LSB}$$

$$8 \overline{) 11} \quad r = 3$$

$$8 \overline{) 1} \quad r = 1 \leftarrow \text{MSB}$$

$$94_{10} = 136_8$$



# Decimal $\leftrightarrow$ Hexadecimal Conversion



The Process: Successive Division

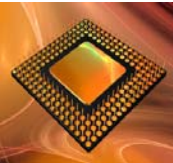
- Divide number by **16**; R is the LSB of the **hex** number
- While Q is 0
- Using the Q as the decimal number.
- New remainder is MSB of the **hex** number.

$$\begin{array}{r} 5 \\ 16 \overline{) 94} \quad r = E \leftarrow \text{LSB} \\ 0 \\ 16 \overline{) 5} \quad r = 5 \leftarrow \text{MSB} \end{array}$$

$$94_{10} = 5E_{16}$$

$$\begin{array}{l} 1A2B3C_{16} \\ 1 \times 16^5 = 1048576 \\ 10 \times 16^4 = 655360 \\ 2 \times 16^3 = 8192 \\ 11 \times 16^2 = 2816 \\ 3 \times 16^1 = 48 \\ 12 \times 16^0 = 12 \\ \hline 1048576 + 655360 + 8192 + 2816 + 48 + 12 \end{array}$$





## Example: Hex $\rightarrow$ Octal

**Example:**

Convert the hexadecimal number 5AH into its octal equivalent.

**Solution:**

First convert the hexadecimal number into its decimal equivalent then convert the decimal number into its octal equivalent.

$$\begin{array}{cc} \mathbf{5} & \mathbf{A} \\ \mathbf{16^1} & \mathbf{16^0} \end{array}$$

$$\begin{array}{cc} \mathbf{16} & \mathbf{1} \end{array}$$

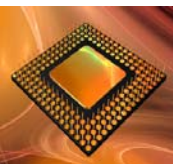
$$\mathbf{80 + 10 = 90_{10}}$$

$$\therefore \mathbf{5A_H = 132_8}$$

$$\begin{array}{r} 11 \\ 8 \overline{) 90} \end{array} \quad r = 2 \leftarrow \text{LSB}$$

$$\begin{array}{r} 1 \\ 8 \overline{) 11} \end{array} \quad r = 3$$

$$\begin{array}{r} 0 \\ 8 \overline{) 1} \end{array} \quad r = 1 \leftarrow \text{MSB}$$



## Example: Octal $\rightarrow$ Binary

**Example:**

Convert the octal number  $132_8$  into its binary equivalent.

**Solution:**

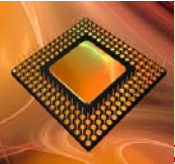
First convert the octal number into its decimal equivalent, then convert the decimal number into its binary equivalent.

<b>1</b>	<b>3</b>	<b>2</b>	
$8^2$	$8^1$	$8^0$	
64	8	1	
<b>64 + 24 + 2 = 90<sub>10</sub></b>			

$2 \overline{) 90}$	$r=0 \leftarrow \text{LSB}$
$2 \overline{) 45}$	$r=1$
$2 \overline{) 22}$	$r=0$
$2 \overline{) 11}$	$r=1$
$2 \overline{) 5}$	$r=1$
$2 \overline{) 2}$	$r=0$
$2 \overline{) 1}$	$r=1 \leftarrow \text{MSB}$

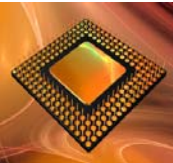
**$132_8 = 1011010_2$**



# Binary $\leftrightarrow$ Octal $\leftrightarrow$ Hex Shortcut



- Relation
- Binary, octal, and hex number systems
- All powers of two
- Exploit (This Relation)
- Make conversion easier.

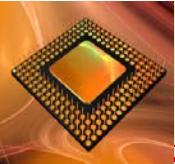


# Substitution Code

Convert  $010101101010111001101010_2$  to hex using the 4-bit substitution code :

5	6	A	E	6	A
0101	0110	1010	1110	0110	1010

**56AE6A<sub>16</sub>**



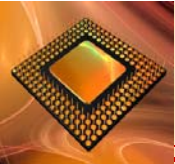
# Substitution Code



Substitution code can also be used to convert binary to octal by using 3-bit groupings:

2	5	5	2	7	1	5	2
010	101	101	010	111	001	101	010

25527152<sub>8</sub>



## Other Representation



### Signed & Unsigned Number

- **Signed number last bit (one MSB) is signed bit**

Assume: 8 bit number

Unsigned 12 : 0000 1100

Signed +12 : **0**000 1100

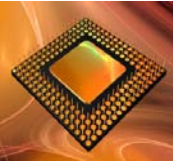
Signed -12 : **1**000 1100

- **Complement number**

Unsigned binary 12 = 0000**11**00

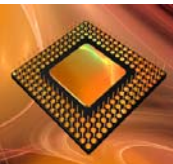
1's Complement of 12 = 1111 **00**11





Example # 1: Convert  $147.8_8$  to decimal.

$$\begin{aligned} 147.3_8 &= 1 \times 8^2 + 4 \times 8^1 + 7 \times 8^0 + 3 \times 8^{-1} = 64 + 32 + 7 + \\ &= 103.375_{10} \end{aligned}$$



## Example # 2:

Convert  $53_{10}$  to binary.

$$\begin{array}{rcl} 2 \overline{)53} & & \\ 2 \overline{)26} & \text{rem.} = 1 = a_0 & \\ 2 \overline{)13} & \text{rem.} = 0 = a_1 & \\ 2 \overline{)6} & \text{rem.} = 1 = a_2 & 53_{10} = 110101_2 \\ 2 \overline{)3} & \text{rem.} = 0 = a_3 & \\ 2 \overline{)1} & \text{rem.} = 1 = a_4 & \\ 0 & \text{rem.} = 1 = a_5 & \end{array}$$



**Thank You**