Jiangxi University of Science and Technology

DIGITAL SYSTEM DESIGN

Lecture0101 introduction / History

History

Digital Systems, Computers, and Beyond

Information Representation

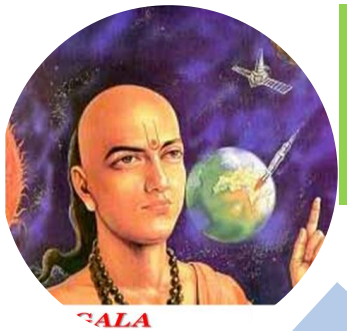
Number Systems [binary, octal and hexadecimal]

Overview on chapter 1

- **History**
- **Digital Systems, Computers, and Beyond**
- **Information Representation**
- **Number Systems** [binary, octal and hexadecimal]
- **Arithmetic Operations**
- **Base Conversion**
- **Decimal Codes** [BCD (binary coded decimal)]
- **Alphanumeric Codes**
- **Parity Bit**
- **Gray Codes**

History

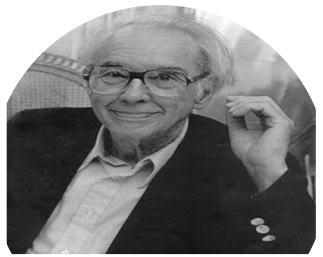
Pingala



Acharya **Pingala** *piṅgala*) (c. 3rd/2nd century BCE)was an ancient Indian mathematician who authored the ***Chandaḥśāstra*** (also called *Pingala-sutras*),

In 1701, Gottfried Wilhelm Leibniz, the person who co-invented Calculus wrote a paper essay D'une Nouvelle Science Des Nombres about his invention. The paper was submitted to the Paris Academy. But, it took another twenty years for the discovery to happen just like it took a few hundred years to develop a binary convertor. According to the available literature on the subject, 1796 was the first time a binary arithmetic entry was reported. So, we can say that the binary number system was born just before the start of the 19th century.

In 1854, British mathematician George Boole published a landmark paper detailing an algebraic system of logic that would become known as Boolean algebra. His logical calculus was to become instrumental in the design of digital electronic circuitry.

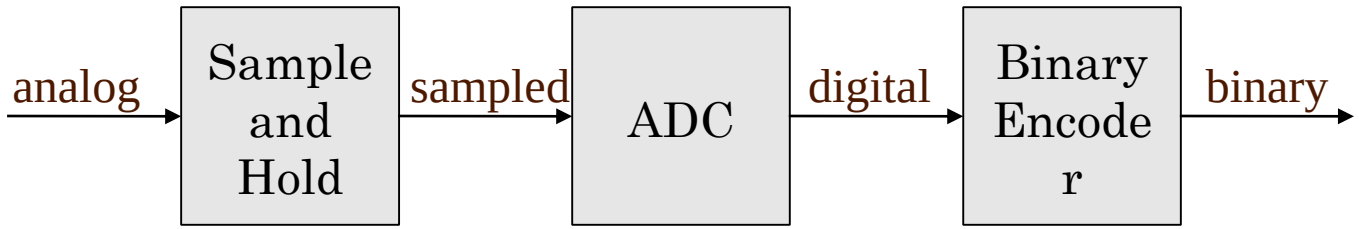


American physicist and inventor.
The 1973 decision of the patent suit Honeywell v. Sperry Rand named him the inventor of the first automatic electronic digital computer.

His special-purpose machine has come to be called the Atanasoff–Berry Computer.

Processing Physical Quantities

- The analog signal (representing the physical quantity) must be sampled at specific instances in time.
- The sampled values must be digitized.
- The digital value must be encoded in binary.

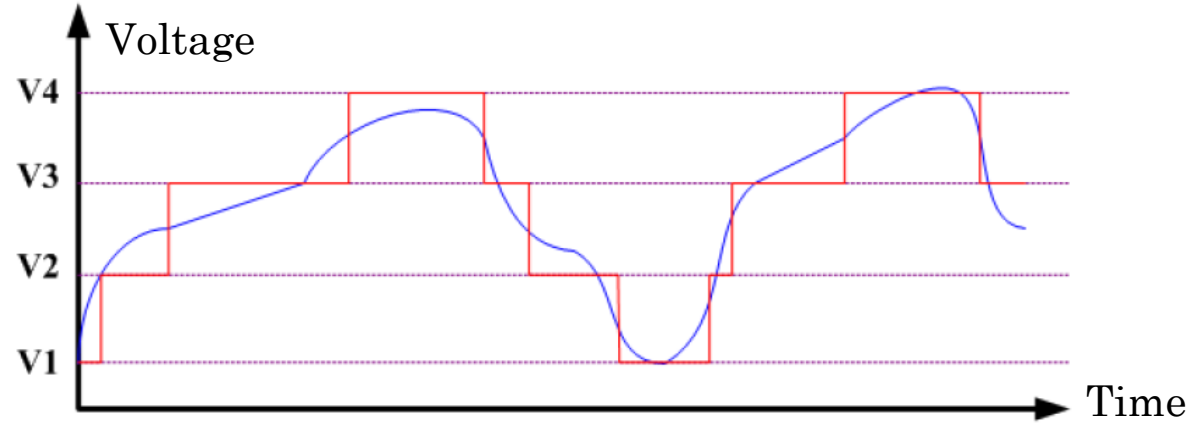


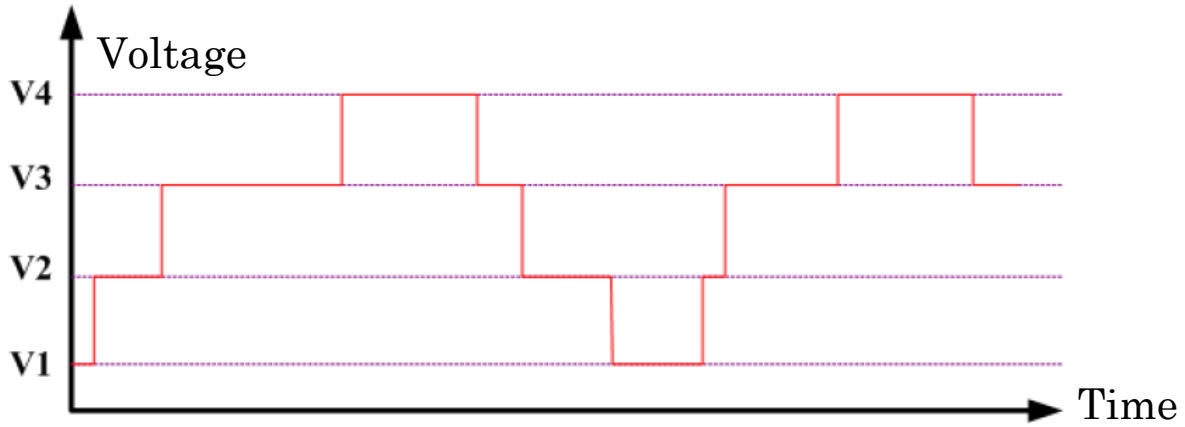
System and signal

- System types
  - Analog➔ can be measures
  - Digital ➔ can be count
- **Binary values are represented abstractly by**
  - **digits 0 and 1**
  - **words (symbols) False (F) and True (T)**
  - **words (symbols) Low (L) and High (H)**
  - **and words On and Off.**
-

Signal Types

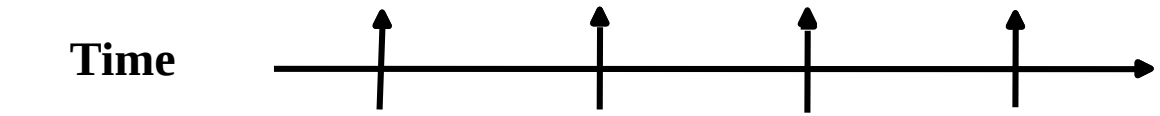
Signal = data
Analog → continues
Digital→ discrete

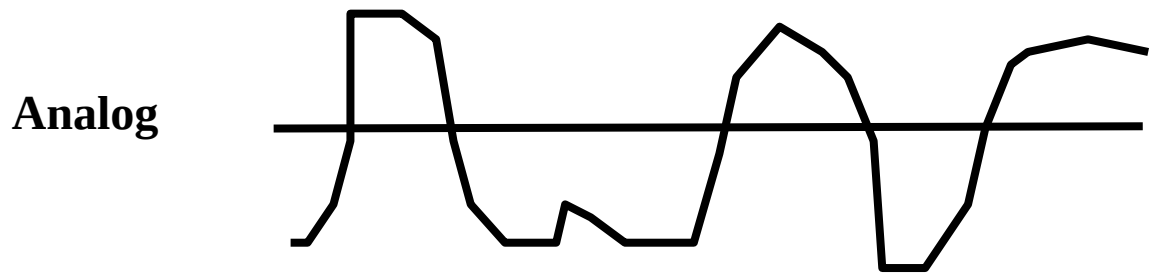




Signal Examples Over Time

Signal = data
Analog → continues
Digital→ discrete



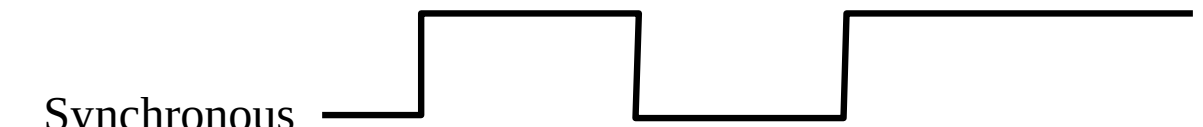


Continuous in value & time

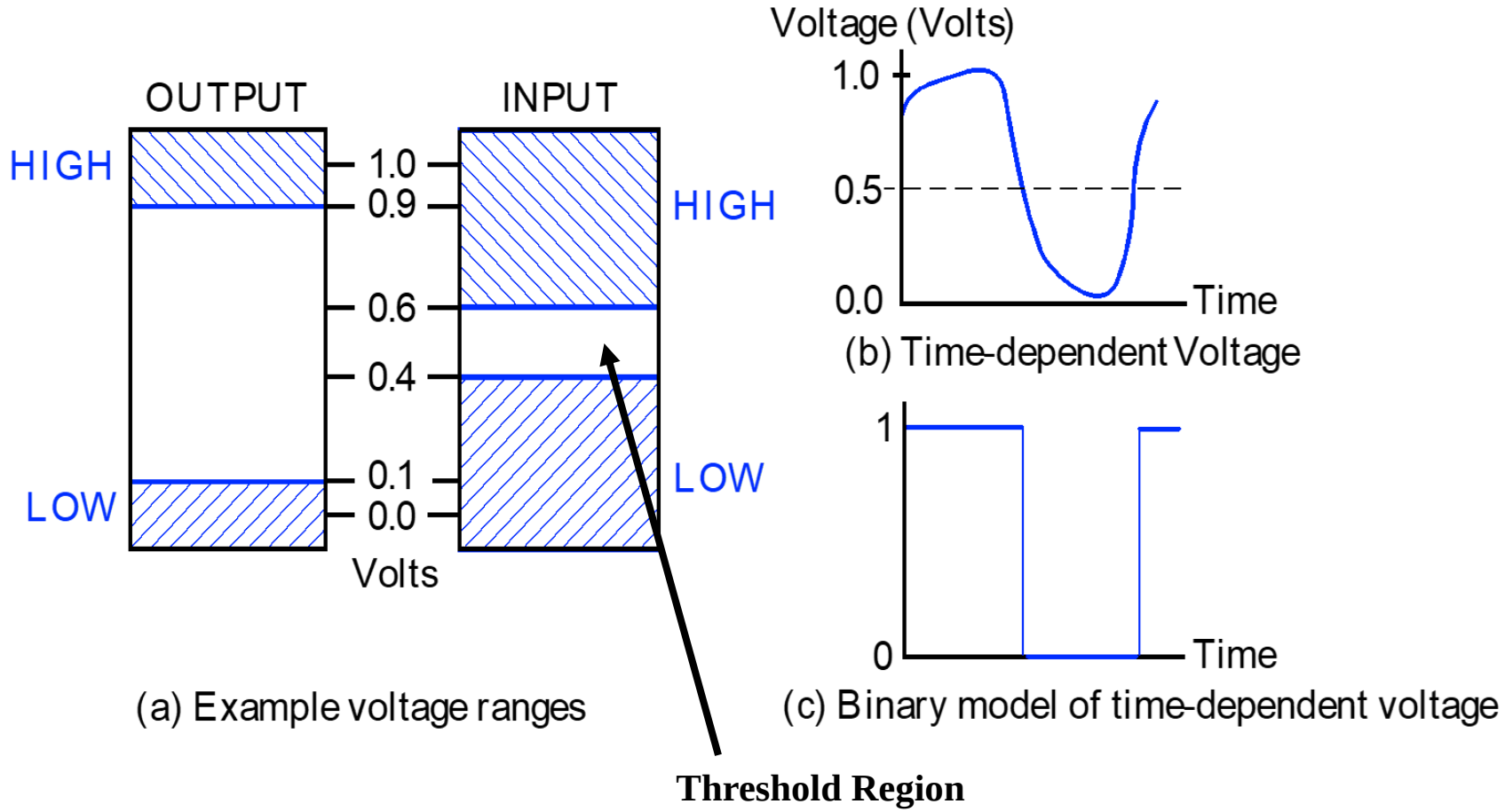
Digital

Discrete in value & continuous in time

Discrete in value & time

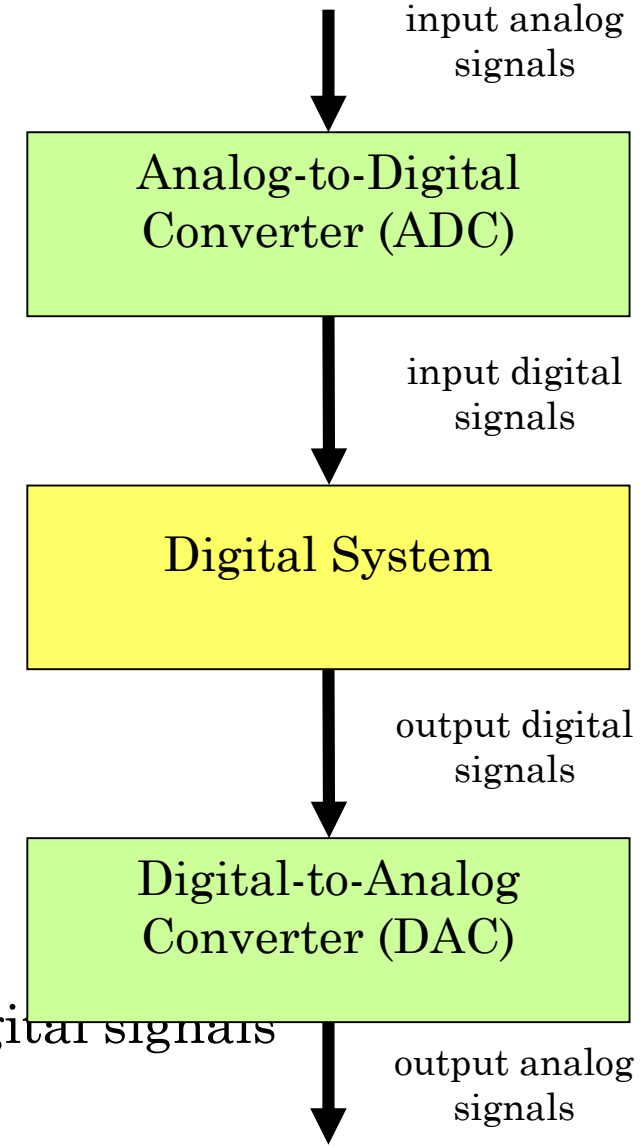


Signal Example – Physical Quantity: Voltage

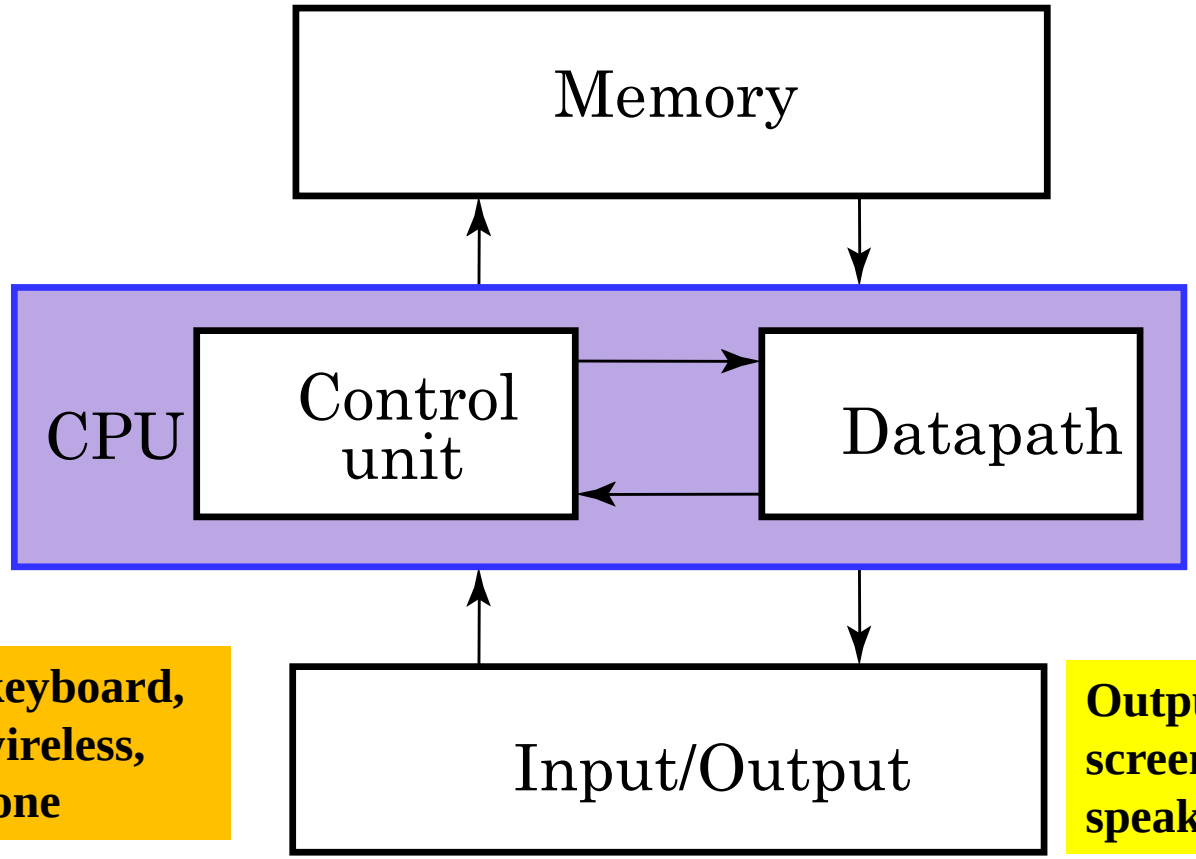


ADC and DAC Converters

- Analog-to-Digital Converter (ADC)
  - Produces digitized version of analog signals
  - Analog input => Digital output
- Digital-to-Analog Converter (DAC)
  - Regenerate analog signal from digital form
  - Digital input => Analog output
- Our focus is on digital systems only
  - Both input and output to a digital system are digital signals



Digital Computer Example



Inputs: keyboard, mouse, wireless, microphone

Outputs: LCD screen, wireless, speakers

Synchronous or Asynchronous?

And Beyond – Embedded Systems

- Computers as integral parts of other products
- Examples of embedded computers
  - Microcomputers
  - Microcontrollers
  - Digital signal processors

- Examples of Embedded Systems Applications
  - Cell phones
  - Automobiles
  - Video games
  - Copiers
  - Dishwashers
  - Flat Panel TVs
  - Global Positioning Systems

Binary Values: Other Physical Quantities

- **What are other physical quantities represent 0 and 1?**

• CPU	Magnetic Field Direction
• Disk	Voltage
• CD	Surface Pits/Light
• Dynamic RAM	Electrical Charge

Number Systems – Representation

- Positive radix, positional number systems
- A number with *radix* r is represented by a string of digits:

$$A_{n-1}A_{n-2}\cdots A_1A_0 . A_{-1}A_{-2}\cdots A_{-m+1}A_{-m}$$

in which $0 \le A_i < r$ and . is the *radix point*.

- The string of digits represents the power series:

$$(\text{Number})_r = \left(\sum_{i=0}^{i=n-1} A_i \cdot r^i\right) + \left(\sum_{j=-m}^{j=-1} A_j \cdot r^j\right)$$

(Integer Portion) + (Fraction Portion)

Number Systems – Examples

		General	Decimal	Binary
Radix (Base)		r	10	2
Digits		0 => r - 1	0 => 9	0 => 1
Powers of Radix	0	r^0	1	1
	1	r^1	10	2
	2	r^2	100	4
	3	r^3	1000	8
	4	r^4	10,000	16
	5	r^5	100,000	32
	-1	r^{-1}	0.1	0.5
	-2	r^{-2}	0.01	0.25
	-3	r^{-3}	0.001	0.125
	-4	r^{-4}	0.0001	0.0625
	-5	r^{-5}	0.00001	0.03125

Special Powers of 2

2^{10} (1024) is Kilo, denoted "K"

2^{20} (1,048,576) is Mega, denoted "M"

2^{30} (1,073, 741,824)is Giga, denoted "G"

2^{40} (1,099,511,627,776) is Tera, denoted “T"

Arithmetic Operations – Binary Arithmetic

- Single Bit Addition with Carry
- Multiple Bit Addition
- Single Bit Subtraction with Borrow
- Multiple Bit Subtraction
- Multiplication
- BCD Addition

Single Bit Binary Addition with Carry

Given two binary digits (X,Y), a carry in (Z) we get the following sum (S) and carry (C):

Carry in (Z) of 0:

Z	0	0	0	0
X	0	0	1	1
+ Y	+ 0	+ 1	+ 0	+ 1
C S	0 0	0 1	0 1	1 0

Carry in (Z) of 1:

Z	1	1	1	1
X	0	0	1	1
+ Y	+ 0	+ 1	+ 0	+ 1
C S	0 1	1 0	1 0	1 1

Multiple Bit Binary Addition

- **Extending this to two multiple bit examples:**

Carries	**0**	**0**
Augend	**01100**	**10110**
Addend	**+10001**	**+10111**
Sum		

- **Note: The 0 is the default Carry-In to the least significant bit.**

Single Bit Binary Subtraction with Borrow

- **Given two binary digits (X,Y), a borrow in (Z) we get the following difference (S) and borrow (B):**
- **Borrow in (Z) of 0:**

Z	0	0	0	0
X	0	0	1	1
- Y	-0	-1	-0	-1
BS	0 0	1 1	0 1	0 0

- **Borrow in (Z) of 1:**

Z	1	1	1	1
X	0	0	1	1
- Y	-0	-1	-0	-1
BS	11	1 0	0 0	1 1

Binary Multiplication

The binary multiplication table is simple:

$0 * 0 = 0 \mid 1 * 0 = 0 \mid 0 * 1 = 0 \mid 1 * 1 = 1$

Extending multiplication to multiple digits:

Multiplicand	1011
Multiplier	x 101
Partial Products	1011
	0000 -
	1011 - -
Product	110111

Weighted number

- If the base of system is R
- We should use R-1 digit to show the numbers(genrally 0 to R-1)

Example:

$A_{(R)} = a_{n-1}\, a_{n-2}\, a_{n-3} \ldots a_1\, a_0 \,.\, a_{-1}\, a_{-2} \ldots a_{-m}$

The value of $A_{(R)}$ is equal to :

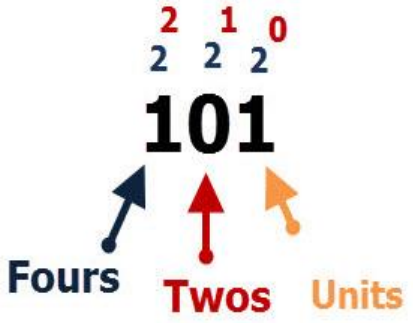
$$\sum_{i=-m}^{n-1} a_i R^i$$

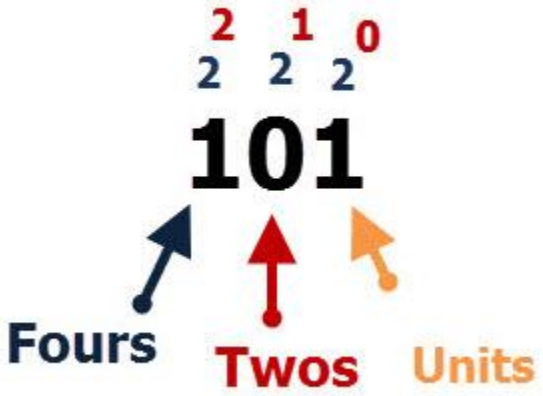
Binary system

- In mathematics and digital electronics, a **binary number** is a number expressed in the **base-2 numeral system** or **binary numeral system**, which uses only two symbols: typically "0" (zero) and "1" (one).
- The base-2 numeral system is a positional notation with a radix of 2. Each digit is referred to as a bit. Because of its straightforward implementation in digital electronic circuitry using logic gates, the binary system is used by almost all modern computers and computer-based devices.

Binary system

- Binary numbers are base 2 numbers, and have only two values – **0 and 1.**
- If we look at a binary number like 101, then we can again assign column values as we did with our decimal number, but this time we use 2, and not 10 as the base.
- So binary **101** binary has 1 in the units column,0 in the 2s column and 1 in the 4s column.
- Again if we work our way from right to left then:
- The 1 is a 1 as it is in the units column but the next 1 is not 1 but 1*4=4





Binary numbers use base 2 and so the columns are

$$2^0 = 1$$
$$2^1 = 2$$
$$2^2 = 4$$
$$2^3 = 8$$

Table 1.1 Powers of Two

Table 1.1
Powers of Two

n	2^n	n	2^n	n	2^n
0	1	8	256	16	65,536
1	2	9	512	17	131,072
2	4	10	1,024 (1K)	18	262,144
3	8	11	2,048	19	524,288
4	16	12	4,096 (4K)	20	1,048,576 (1M)
5	32	13	8,192	21	2,097,152
6	64	14	16,384	22	4,194,304
7	128	15	32,768	23	8,388,608

Bases

- Important bases are
  - 2_binary _B
  - 8_Octal_O
  - 10_decimal_D
  - 16_hexadecimal_H

- Numbers in base 16(hexa)
- 0 1 2 3 4 5 6 7 8 9 A B C D E F

Table 1.2
Numbers with Different Bases

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
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

Decimal Number systems

- Its include number 0 – 9
- And the location of number in the weighted system gives the values

$$\begin{array}{cccccl} \ldots & 10^4 & 10^3 & 10^2 & 10^1 & 10^0 \\ & & & 1 & 7 & 3 \end{array} \quad = 1\times10^2 + 7\times10^1 + 3\times10^0$$

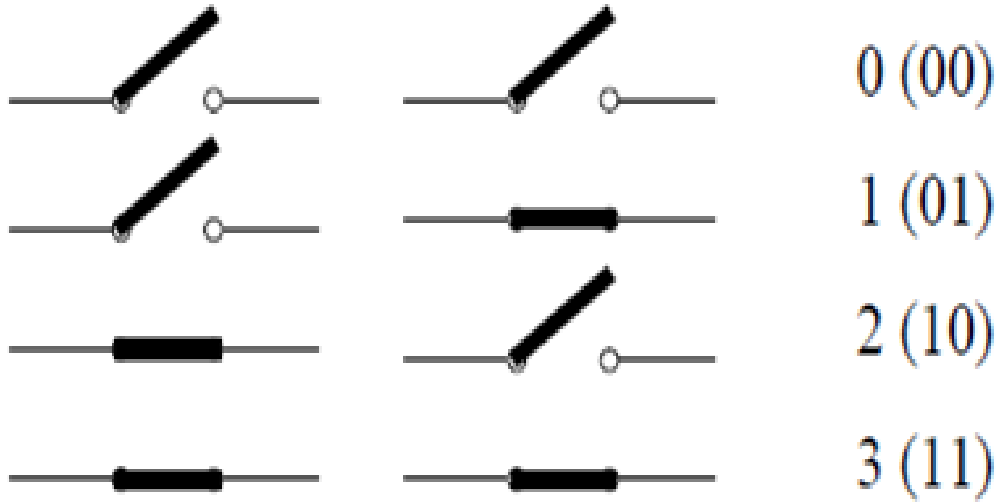
$$= 100 + 70 + 3$$

$$= 173$$

Binary system

- Electrical switch's are present two state of Off and ON which is equal to binary digit and showing the bit status

- In the case of having two bit we can show 4 state



Binary system

- Simple for computer and difficult for human
- (binary digits (bits)) numbers are used instad of decimal numbers
- n bit data canpresent 2n numbers
- In this system the position of number will shows the number weights
- Each 4 bit called one nibble
- Each 8 bit called one byte
- Each two bytes called one word

Why we use the Binary system

1. Simple to show the 0, 1 values for electrical signals
2. Good margins in noise
3. Simple to send
4. Simple to made the binary circuits

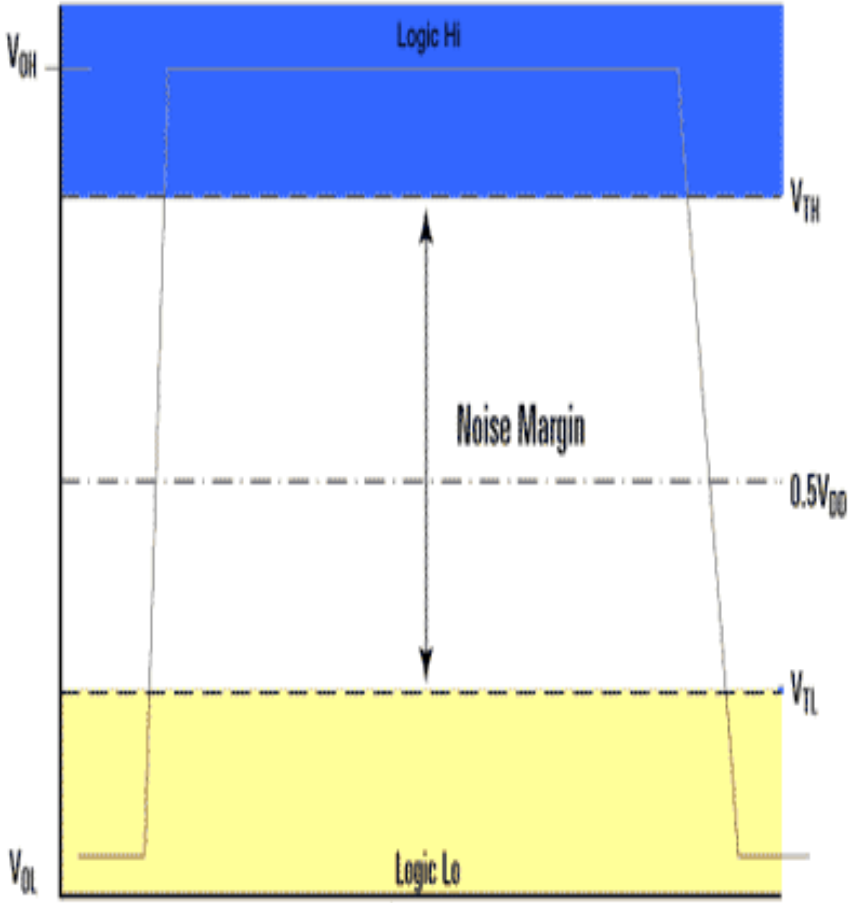
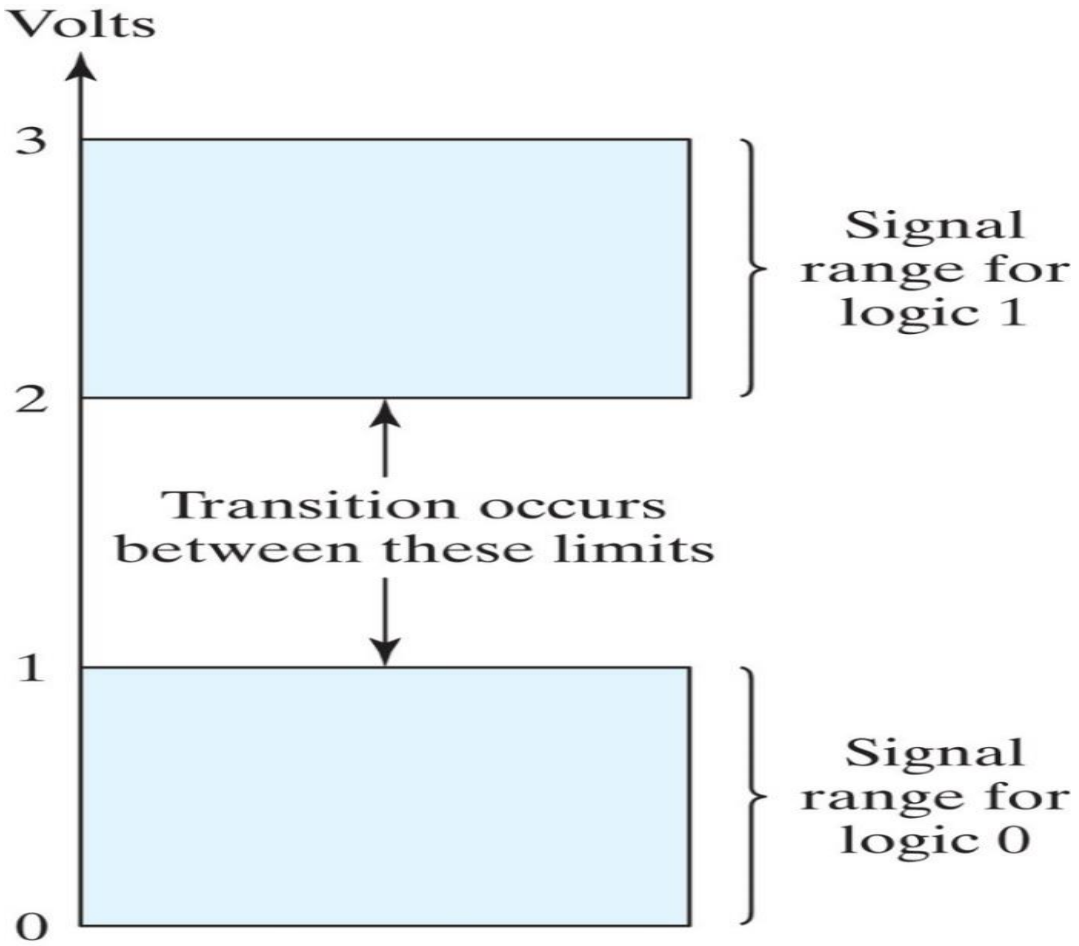


FIGURE 1.3 Signal levels for binary logic values



Binary To Decimal Conversion

- Let's look at a few binary numbers and convert them to decimal
- We start with the three digit binary number **101** (see image above
- The number can be converted to decimal by multiplying out as follows:

  1*1 + 0*2 + 1*4 = 5

- The maximum value we can have with three binary digits is

  111 = decimal 7 calculated as follows-1*1 + 1*2 + 1*4

More Examples:

- 1011 binary = 1*1+1*2+0*4+1*8=11
- 1111 binary = 1*1+1*2+1*4+1*8=15

- **Try It Yourself**
  - 1001 binary = ?
  - 1100 binary = ?

Number peresntation

$$7392 = 7 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$$

$$a_5 a_4 a_3 a_2 a_1 a_0 \cdot a_{-1} a_{-2} a_{-3}$$

$$10^5 a_5 + 10^4 a_4 + 10^3 a_3 + 10^2 a_2 + 10^1 a_1 + 10^0 a_0 + 10^{-1} a_{-1} + 10^{-2} a_{-2} + 10^{-3} a_{-3}$$

$$11010.11 \quad 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 26.75$$

$$(4021.2)_5 = 4 \times 5^3 + 0 \times 5^2 + 2 \times 5^1 + 1 \times 5^0 + 2 \times 5^{-1} = (511.4)_{10}$$

$$(127.4)_8 = 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} = (87.5)_{10}$$

$$(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = (46{,}687)_{10}$$

$$(110101)_2 = 32 + 16 + 4 + 1 = (53)_{10}$$

Converting Binary to Decimal

- To convert to decimal, use decimal arithmetic to form Σ (digit × respective power of 2).
- Example:Convert 11010_2 to N_{10}:

  - Method 1
    - Subtract the largest power of 2 (see slide 14) that gives a positive remainder and record the power.
    - Repeat, subtracting from the prior remainder and recording the power, until the remainder is zero.
    - Place 1's in the positions in the binary result corresponding to the powers recorded; in all other positions place 0's.
  - Example: Convert 625_{10} to N_2

Converting Binary to Decimal

Method 2 : To convert from one base to another:

1) Convert the Integer Part

2) Convert the Fraction Part

3) Join the two results with a radix point

Conversion Details

- **To Convert the Integral Part:**

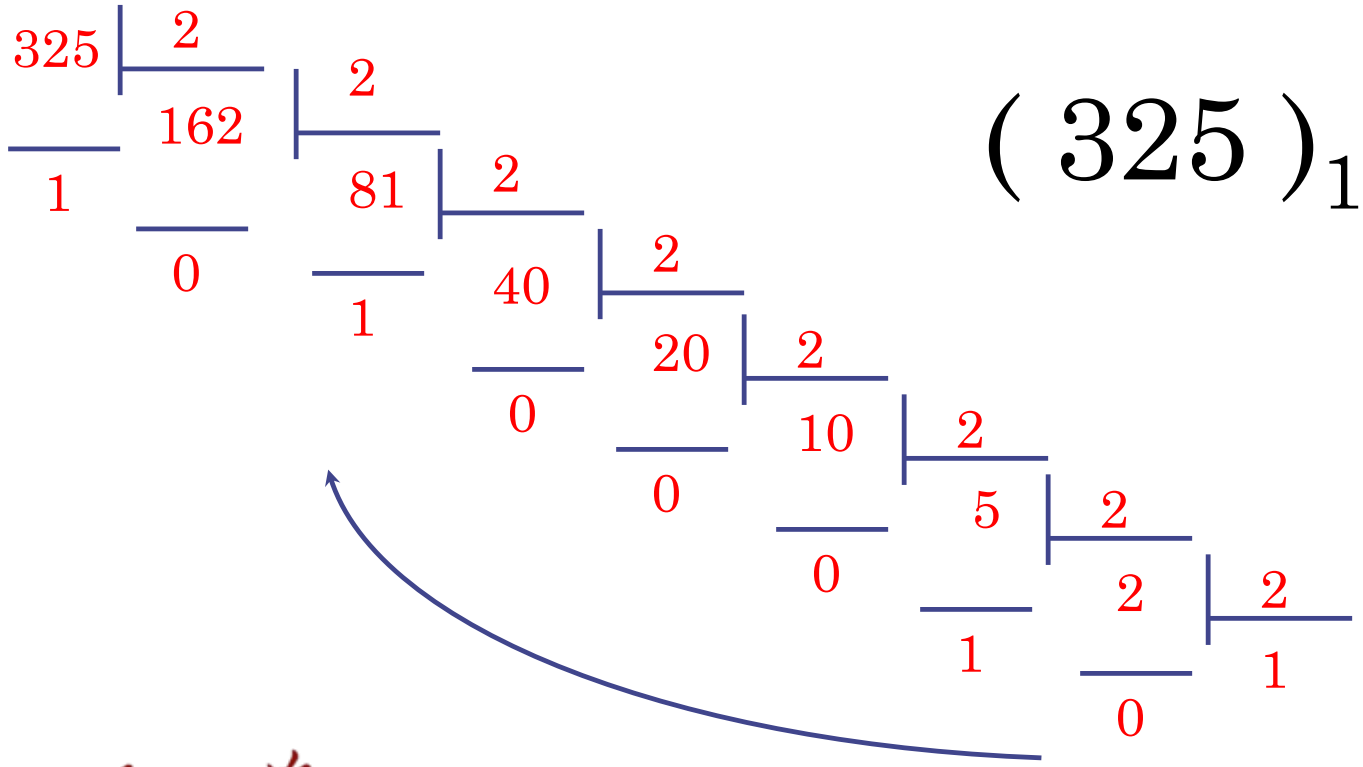
  Repeatedly divide the number by the new radix and save the remainders. The digits for the new radix are the remainders in *reverse order* of their computation. If the new radix is > 10, then convert all remainders > 10 to digits A, B, …

- **To Convert the Fractional Part:**

  Repeatedly multiply the fraction by the new radix and save the integer digits that result. The digits for the new radix are the integer digits in *order* of their computation. If the new radix is > 10, then convert all integers > 10 to digits A, B, …

Example on Number and base

$(325)10 = (?)2$



$$(\ 325\ )_{10} \rightarrow (\ 101000101\ )_2$$

Example on Number and base

$(41.6875)10 = (?)2$

Integer section= 41

41	
20	1
10	0
5	0
2	1
1	0
0	1

Float section= 0.6875

0.6875
x 2
1.3750
x 2
0.7500
x 2
1.5000
x 2
1.0000

$(41)_{10} = (101001)_2$

$(0.6875)_{10} = (0.1011)_2$

$(41.6875)_{10} = (101001.1011)_2$

Example on Number and base

$$(1101.101)_2 = 1\times2^3 + 1\times2^2 + 1\times2^0 + 1\times2^{-1} + 1\times2^{-3}$$
$$= 8 + 4 + 1 + 0.5 + 0.125 = (13.625)_{10}$$

$$(572.6)_8 = 5\times8^2 + 7\times8^1 + 2\times8^0 + 6\times8^{-1}$$
$$= 320 + 56 + 2 + 0.75 = (378.75)_{10}$$

$$(2A.8)_{16} = 2\times16^1 + 10\times16^0 + 8\times16^{-1}$$
$$= 32 + 10 + 0.5 = (42.5)_{10}$$

$$(341.24)_5 = 3\times5^2 + 4\times5^1 + 1\times5^0 + 2\times5^{-1} + 4\times5^{-2}$$
$$= 75 + 20 + 1 + 0.4 + 0.16 = (96.56)_{10}$$

Example on Number and base

(43.3125)10 = (?)2

2	43	
2	21 rem 1	← LSB
2	10 rem 1	
2	5 rem 0	
2	2 rem 1	
2	1 rem 0	
	0 rem 1	← MSB

$(43)_{10} = (101011)_2$

	Carry	
0.3125×2=0.625	0	←MSB
0.625×2=1.25	1	
0.25×2=0.50	0	
0.5×2=1.00	1	←LSB

$(0.3125)_{10} = (.0101)_2$

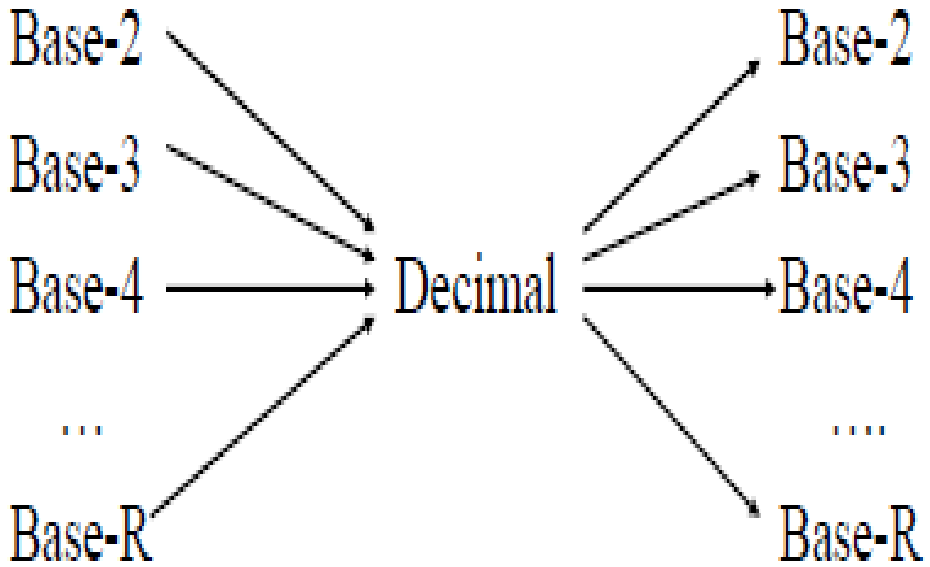
$$(43.3125)_{10} = (101011.0101)_2$$

Example on Number and base

- ✓(1101.101)2 = (?)10 (13.625)10
- ✓ (572.6)8 = (?)10 (378.75)10
- ✓ (2A.8)16 = (?)10 (42.5)10
- ✓ (341.24)5 = (?)10 (96.56)10
- ✓(76)10=(?)2
- ✓(26)10=(?)2

Transfer between Base 2, 8,16

- In the general form the base 10 (decimal) has the intermedia role between the bases
- For the bases like 4,8,16 there is some simpler way



Octal (Hexadecimal) to Binary and Back

- **Octal (Hexadecimal) to Binary:**
  - **Restate the octal (hexadecimal) as three (four) binary digits starting at the radix point and going both ways.**
- **Binary to Octal (Hexadecimal):**
  - **Group the binary digits into three (four) bit groups starting at the radix point and going both ways, padding with zeros as needed in the fractional part.**
  - **Convert each group of three bits to an octal (hexadecimal) digit.**

Octal to Hexadecimal via Binary

- Convert octal to binary.
- Use groups of four bits and convert as above to hexadecimal digits.
- Example: Octal to Binary to Hexadecimal

$$6\ \ 3\ \ 5\ .\ 1\ \ 7\ \ 7_{8}$$

- Why do these conversions work?

Change/transfer betwwen BASE 2-8- 16

- $(1010111101100011)_2$
- $(1010111101100011)_2 = (127543)_8 = (AF63)_{16}$

	1	2	7	5	4	3
Octal						
Binary	1 0 1 0 1 1 1 1 0 1 1 0 0 0 1 1					
Hexa		A	F	6	3	

Transfer between Base 2, 8,16

Binary → Octal: Partition in groups of 3

$(10\ 111\ 011\ 001\ .\ 101\ 110)_2 = (2731.56)_8$

Octal → Binary: reverse

$(2731.56)_8 = (10\ 111\ 011\ 001\ .\ 101\ 110)_2$

Binary → Hexadecimal: Partition in groups of 4

$(101\ 1101\ 1001\ .\ 1011\ 1000)_2 = (5D9.B8)_{16}$

Hexadecimal → Binary: reverse

$(5D9.B8)_{16} = (101\ 1101\ 1001\ .\ 1011\ 1000)_2$

Example on Number and base

(5762)8 = (?)2

(1011011)2 = (?)8

(39)10 = (?)2

(10011)2 = (?)10

(63)10 = (?)8

(22.73)10 = (?)2

(10110.1011)2 = (?)10

(10 0100 1001)2 = (?)16

(A3C9E)16 = (?)2

(1011101101)2 =(?)8 = (?)16