INTRODUCTION TO ICT

Lesson 2

Representation of Data in computer

Content

I. Numeral systems

II. Data representation in computer

III. Information unit

IV. Representation of integer number

V. Representation of real number

VI. Representation of character

I. Numeral systems

- **Numeral system**:
  - Is a set of symbols and rules, using which we can represent and determine the value of numbers.
  - Each numeral system uses a finite set of characters/symbols/digits. The number of characters/symbols/digits is called the base or radix of the numeral system, normally denoted by b
  - E.g.: In the decimal system, we use ten digits: 0,1,2,…,9, hence b =10

I. Numeral systems

- Mathematically, we can represent a number in any numeral system
- In computer science, we usually using following counting systems:
  - Decimal system
  - Binary system
  - Octal system
  - Hexadecimal system

Decimal system

- The decimal system (with b = 10) using 10 digits:

  0, 1, 2, 3, 4, 5, 6, 7, 8, 9

- In the decimal system, using n digits we can represent 10^n different values, ranging from 0 to 10^n-1:

  00...000 = 0

  ....

  99...999 = 10^n-1

Decimal system

- Supposing that a number A which is represented in the decimal system as:

  $A = a_n\, a_{n-1} \ldots a_1\, a_0 \,.\, a_{-1}\, a_{-2} \ldots a_{-m}$

  → The value of A is determined as:

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

$$A = \sum_{i=-m}^{n} a_i 10^i$$

Decimal system

- E.g. 1: The value of the number 5246 is determined as:

  $5246 = 5 \times 10^{3} + 2 \times 10^{2} + 4 \times 10^{1} + 6 \times 10^{0}$

- E.g. 2: The value of the number 254.68 is determined as:

  $254.68 = 2 \times 10^{2} + 5 \times 10^{1} + 4 \times 10^{0} + 6 \times 10^{-1} + 8 \times 10^{-2}$

b-base system

- In general, a b-base numeral system (b >=2, integer):
  - Using b digits, ranging from 0 to b-1
  - A number $N_{(b)}$ in the b-base numeral system is represented as:

    $N_{(b)}=a_{n-1}a_{n-2...}a_1a_0.a_{-1}a_{-2}...a_{-m}$
  - In the above representation, n digits is used to represent integer part and m digits for fraction

b-base system

– The value of number $N_{(b)}$ is determined as:

$$N_{(b)} = a_n.b^n + a_{n-1}.b^{n-1} + a_{n-2}.b^{n-2} + \ldots + a_1.b^1 + a_0.b^0 + a_{-1}.b^{-1} + a_{-2}.b^{-2} + \ldots + a_{-m}.b^{-m}$$

hay là:

$$N_{(b)} = \sum_{i=-m}^{n} a_i.b^i$$

The binary system

- Base b = 2
- Using 2 digits: 0 and 1
- Binary digit (0 or 1) is called a ***bit*** (**bi**nary digi**t)**

  E.g.: bit 0, bit 1
- Bit is the smallest information unit

The binary system

- Using n bits we can represent 2^n different value, ranging from 0 to 2^n -1:

  00...000 (2)= 0

  ...

  11...111 (2)= 2^n - 1

- E.g.: Using 2 bits we can represent 4 values: 00 (=0); 01 (=1); 10 (=2); 11 (=3)

The binary system

- Supposing that a number A which is represented in the binary system as:

  A = a_n a_{n-1} … a_1 a_0 . a_{-1} a_{-2} … a_{-m}

,in which a_i is binary digit (0,1), then value of A is determined as:

$$A = a_n 2^n + a_{n-1} 2^{n-1} + \ldots + a_1 2^1 + a_0 2^0 + a_{-1} 2^{-1} + a_{-2} 2^{-2} + \ldots + a_{-m} 2^{-m}$$

$$A = \sum_{i=-m}^{n} a_i 2^i$$

The binary system

- E.g.: The value of the binary number 1101001.1011 is determined as:

$1101001.1011_{(2)} = 2^6 + 2^5 + 2^3 + 2^0 + 2^{-1} + 2^{-3} + 2^{-4}$

$= 64 + 32 + 8 + 1 + 0.5 + 0.125 + 0.0625$

$= 105.6875_{(10)}$

Plus/Minis in the binary system

- Plus:

  1+0=0+1=1;

  0+0=0;

  1+1=10;

- Minus

  1-1=0;

  0-0=0;

  1-0=1;

  0-1=1; (borrow 1)

Example: Plus

```
    1  0   1
 + 1 1    1
    ---------
  11 0    0
```

Example: Minus

```
        1  1  0   0
-          1  1   1
  --------------------
        0 1   0    1
```

The octal system

- Base b = 8
- Using 8 digits: 0,1,2,3,4,5,6,7
- With n digits we can represent 8^n different values, ranging from 0 to 8^n -1 as:

  00...000 = 0

  ...

  77...777 = 8^n -1

The octal system

- Supposing that an A number which is represented in the octal system as:

$$A = a_n\, a_{n-1} \ldots a_1\, a_0\, .\, a_{-1}\, a_{-2} \ldots a_{-m}$$

with a_i is digit in the octal system, then value of A is determined as:

$$N_{(b)} = a_n.b^n + a_{n-1}.b^{n-1} + a_{n-2}.b^{n-2} + \ldots + a_1.b^1 + a_0.b^0 + a_{-1}.b^{-1} + a_{-2}.b^{-2} + \ldots + a_{-m}.b^{-m}$$

hay là:

$$N_{(b)} = \sum_{i=-m}^{n} a_i.b^i$$

with b = 8.

The octal system

- E.g.: Determine the value of

  $235\,.\,64_{(8)}$:

  $235\,.\,64_{(8)} = \mathbf{2}\text{x}8^{2} + \mathbf{3}\text{x}8^{1} + \mathbf{5}\text{x}8^{0} + \mathbf{6}\text{x}8^{-1} + \mathbf{4}\text{x}8^{-2} =$

  157. 8125(10)

The hexadecimal system

- Base b = 16
- Using 16 digits and characters:

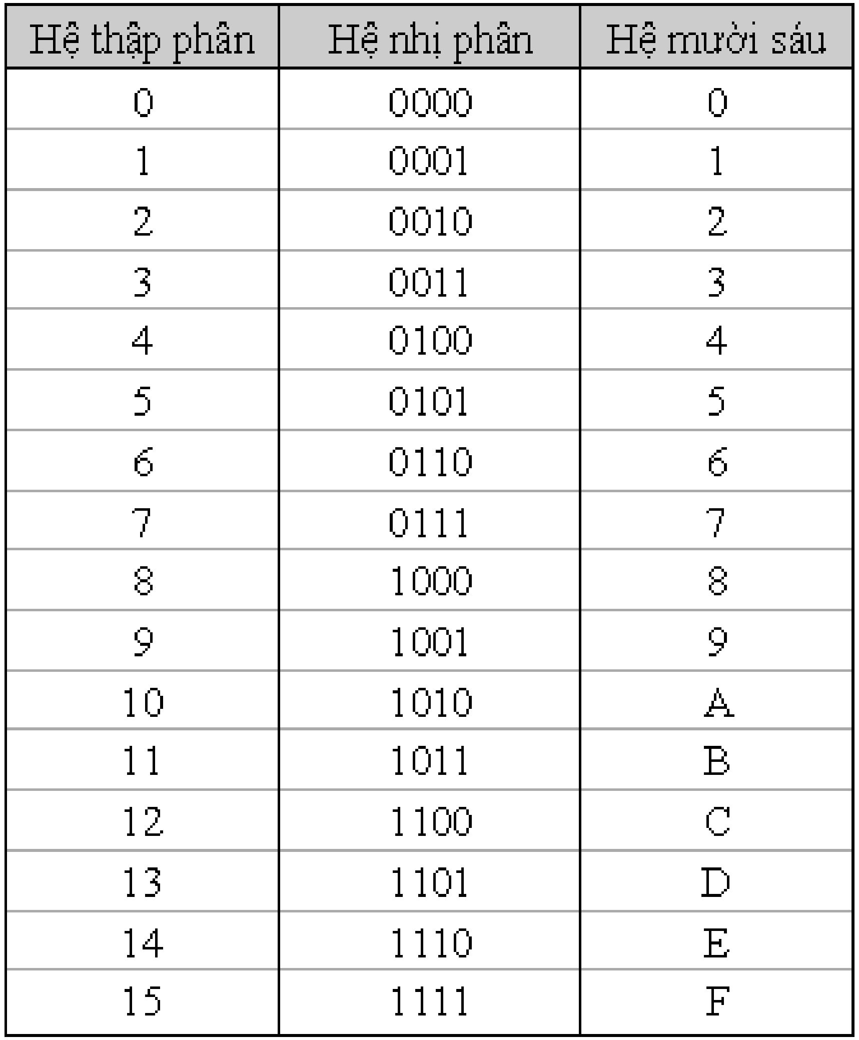
  0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

  in which characters:

  A, B, C, D, E, F

  represent respected number in decimal system

  10, 11, 12, 13, 14, 15.



Hệ thập phân	Hệ nhị phân	Hệ mười sáu
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

The hexadecimal system

- Supposing that an A number which is represented in hexadecimal system as:

  $A = a_n \, a_{n-1} \ldots a_1 \, a_0 \, . \, a_{-1} \, a_{-2} \ldots a_{-m}$

  with a_i is digit in hexadecimal system, the value of A is determined as:

$$A = a_n 16^n + a_{n-1} 16^{n-1} + \ldots + a_1 16^1 + a_0 16^0 + a_{-1} 16^{-1} + a_{-2} 16^{-2} + \ldots + a_{-m} 16^{-m}$$

$$A = \sum_{i=-m}^{n} a_i 16^i$$

The hexadecimal system

- E.g.: Determine value of $34F5C.12D_{(16)}$:

$34F5C.12D_{(16)}$ =

$3x16^4 + 4x16^3 + 15x16^2 + 5x16^1 + 12x16^0$

+? = 216294(10) +?

?=1/16 + 2/16/16 +13/16/16/16=0.07348..

Convert a number from decimal system to other numeral system

- In general a number N in the decimal system ($N_{(10)}$) consists of 2 part: Integer and Fraction.
- Converting a number from decimal system to other numeral system including 2 steps:
  - Convert integer part from decimal system to b-base system
  - Convert fraction part from decimal system to b-base system

Convert a number from decimal system to other numeral system

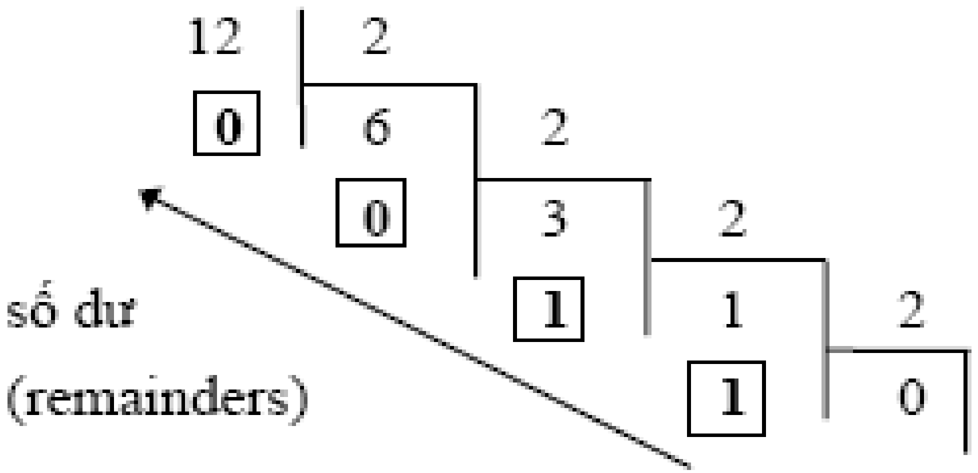
- Convert integer part:
  - **Step 1:** Divide the integer part of $N_{(10)}$ by b, we obtain T_1, remainder is $d_{1.}$
  - **Step 2:** if T_1 <> 0, Divide T_1 by b, we obtain T_2, remainder is d_2

    ….
  - **Step n:** T_n = 0, remainder is d_n => stop
  - Result number N(b) obtained by remainders in above steps as:

    $N_{(10)} = d_n d_{n-1} \ldots d_{1\ (b)}$

Convert a number from decimal system to other numeral system

– E.g.: Convert integer part of number $12.6875_{(10)}$ to binary system:



Kết quả: $12_{(10)} = 1100_{(2)}$

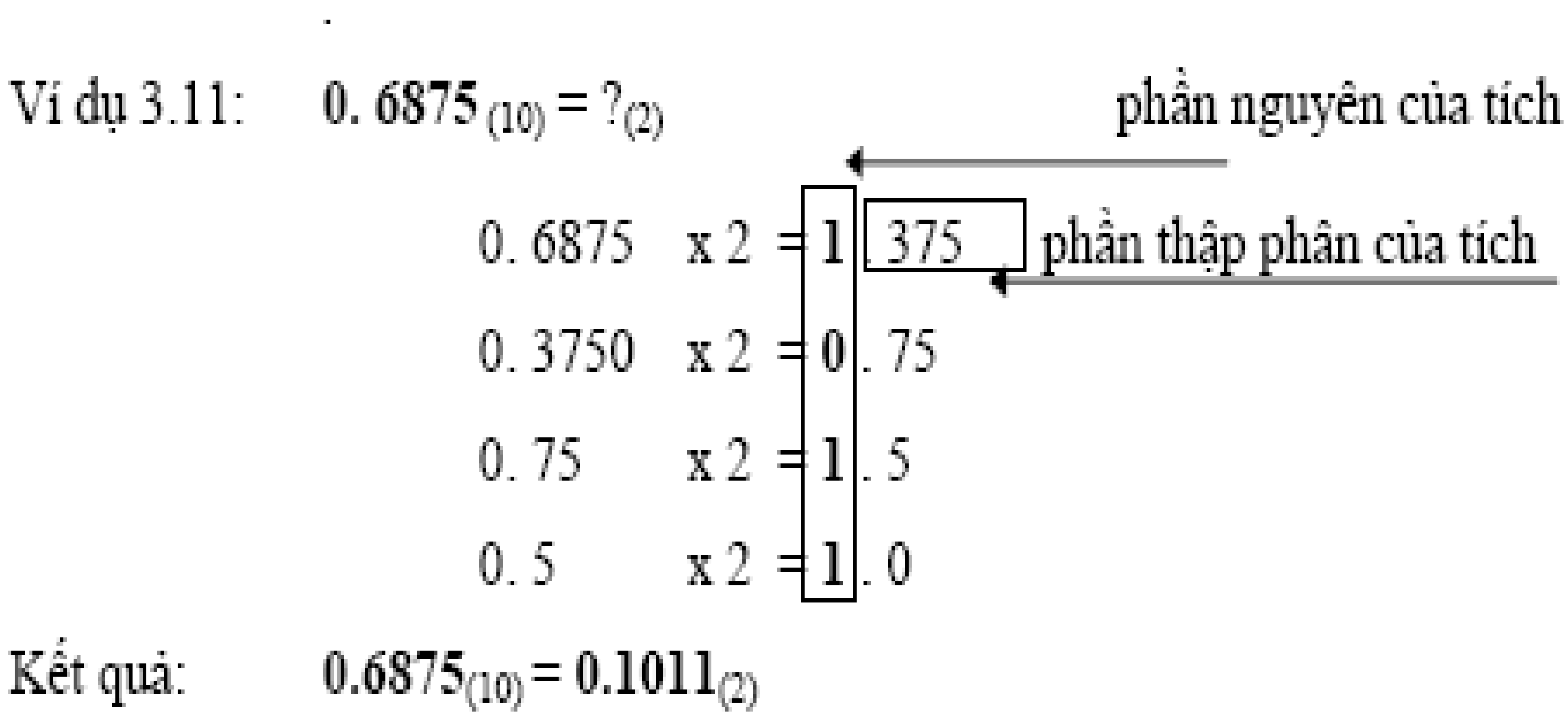
Convert a number from decimal system to other numeral system

- Convert fraction part of a number in decimal system to other system:
  - Step 1: Multiple fraction part of $N_{(10)}$ with b, we obtain a number in form of $x_1.y_1$ (in which x in integer, y is fraction)
  - Step 2: If y_1 <> 0, multipe $0.y_1$ with b, we obtain $x_2.y_2$
  - Step n: If y_{n-1} <> 0, multiple $0.y_{n-1}$ with b, we obtain $x_n.0$, => stop
  - Result number is:

    $\mathbf{0,x_1x_2\ldots x_n}$

Convert a number from decimal system to other numeral system

- E.g.: Convert fraction part of number $12.6875_{(10)}$ to binary system:



Ví dụ 3.11: $0.6875_{(10)} = ?_{(2)}$

phần nguyên của tích

$0.6875 \times 2 = 1.375$ phần thập phân của tích

$0.3750 \times 2 = 0.75$

$0.75 \times 2 = 1.5$

$0.5 \times 2 = 1.0$

Kết quả: $0.6875_{(10)} = 0.1011_{(2)}$

Examples

1. Decimal → Binary 124.75 = ?

    1111100.11

2. Decimal → Binary 65.125 = ?

    1000001.001

3. Binary → Hexa: 1011 1110 0110 = ?

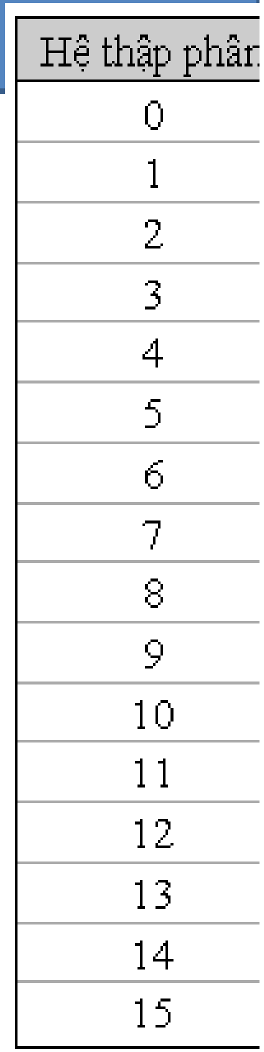
BE6

4. Hexa → Binary: 3E8 = ?

0011 1110 1000

5. Hexa → Decimal: 3A8C = ?

= 14988



Hệ thập phân
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

- 124.75
- 124 /2 = 62 , 0
- 62 /2 = 31 , 0
- 31 / 2 = 15 , 1
- 15 / 2 = 7 , 1
- 7 /2 = 3 , 1
- 3 / 2 = 1 , 1
- 1 / 2 = 0 , 1
- 124 = 1111100

- 0.75 x2 = 1 , 5
- 0.5 x 2 = 1 , 0
- -.75 = 11
- 124.75 = 1111100.11

Represent data in computer

- All data using in computer must be encoded into binary numbers
- Data in computer is classified into 2 types:
  - Basic data
  - Structured data

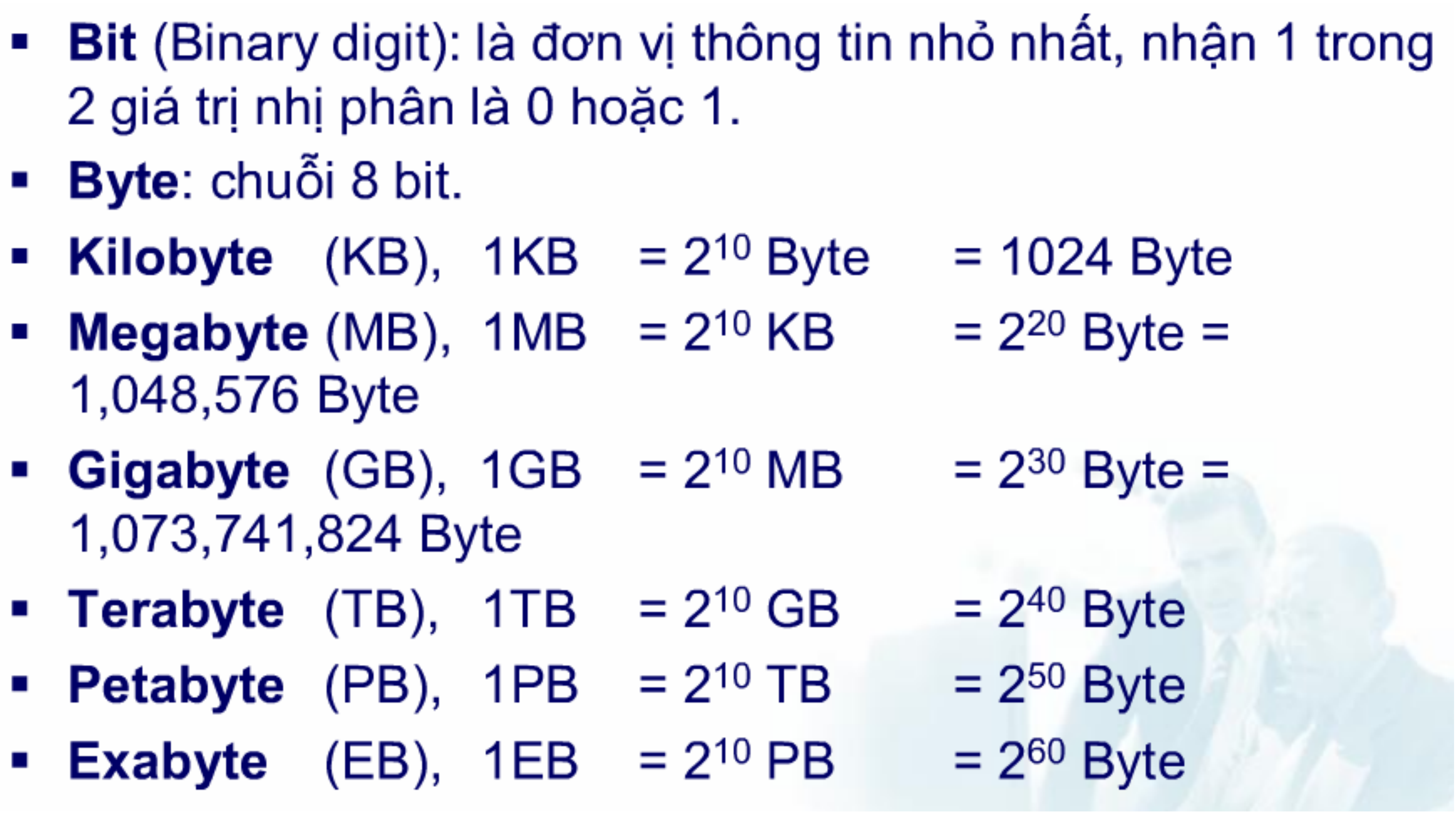
Basic data

- Integer
  - Unsigned integer: Using binary.
  - Signed integer: Using the 2's complement encoding.
- Real:
  - Using floating point number
- Character:
  - Using character encode: ASCII, Unicode, …

Structured data

- Is a set of basic data which is structured in a specific manner.
- E.g.: Array, string, record, …
- Study more in other courses like Programming language.

Information Unit



- **Bit** (Binary digit): là đơn vị thông tin nhỏ nhất, nhận 1 trong 2 giá trị nhị phân là 0 hoặc 1.
- **Byte**: chuỗi 8 bit.
- **Kilobyte** (KB), 1KB = 2^{10} Byte = 1024 Byte
- **Megabyte** (MB), 1MB = 2^{10} KB = 2^{20} Byte = 1,048,576 Byte
- **Gigabyte** (GB), 1GB = 2^{10} MB = 2^{30} Byte = 1,073,741,824 Byte
- **Terabyte** (TB), 1TB = 2^{10} GB = 2^{40} Byte
- **Petabyte** (PB), 1PB = 2^{10} TB = 2^{50} Byte
- **Exabyte** (EB), 1EB = 2^{10} PB = 2^{60} Byte

Represent data in computer

1. Representation of integer number
2. Representation of real number
3. Representation of character

Representation of integer number

- Using a set of bits.
- For signed integer, we use the first bit (Most significant bit) to represent "-" sign. We call this bit is signed bit.

Representation of unsigned integer

- Supposing that a number A is represented using n bits as:

$$a_{n-1}a_{n-2}\ldots a_3a_2a_1a_0$$

then, value of A is determined as:

$$A = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \ldots + a_1 2^1 + a_0 2^0$$

$$A = \sum_{i=0}^{n-1} a_i 2^i$$

- Using n bit to represent unsigned integer, we can represent values from 0 to 2^n-1

Examples

- E.g. 1: Representing unsigned integer A and B numbers using 8 bits:

A = 45 B = 156

$A = 45 = 32 + 8 + 4 + 1 = 2^5 + 2^3 + 2^2 + 2^0$

$\rightarrow$ A = 0010 1101

$B = 156 = 128 + 16 + 8 + 4 = 2^7 + 2^4 + 2^3 + 2^2$

$\rightarrow$ B = 1001 1100

Examples

- E.g. 2: Given representation of unsigned integer numbers X and Y, determine values of X and Y:

  X = 0010 1011

  Y = 1001 0110

  $X = 0010\ 1011 = 2^5 + 2^3 + 2^1 + 2^0$

  $= 32 + 8 + 2 + 1 = 43$

  $Y = 1001\ 0110 = 2^7 + 2^4 + 2^2 + 2^1$

  $= 128 + 16 + 4 + 2 = 150$

Representation of signed integer

- Given representation of signed integer A as:

$$a_{n-1}a_{n-2}\ldots a_2a_1a_0$$

then, value of A is determined as:

$$A = -a_{n-1}2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$

- Using n bits, values of A is in: $[-2^{n-1}, 2^{n-1}-1]$

10000…000

……….

01111…111

Representation of signed integer

- Given representation of signed integer A as:

$$a_{n-1}a_{n-2}\ldots a_2a_1a_0$$

- If a_{n-1}:= 0:
  - A is positive number
  - $a_{n-2}\ldots a_2a_1a_0$ represent value of A

$$A = \sum_{i=0}^{n-2} a_i 2^i$$

  - Values of A is in: $[0, 2^{n-1}-1]$

Representation of signed integer

- Given representation of signed integer A as:

$$a_{n-1}a_{n-2}\ldots a_2a_1a_0$$

- If a_{n-1}:= 1:
  - A is negative number
  - Value of A is determined as:

$$A = -2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$$

– Values of A is in: $[-2^{n-1}, -1]$

Representation of signed integer

- E.g.: Determine value of signed integers A and B:

A = 0101 0110

B = 1101 0010

$A = 2^6 + 2^4 + 2^2 + 2^1 = 64 + 16 + 4 + 2 = +86$

$B = -2^7 + 2^6 + 2^4 + 2^1 =$

$= -128 + 64 + 16 + 2 = -46$

Representation of signed integer

Convert a signed integer number (+/-) from decimal system to binary system

Representation of signed integer

- Using the 2’s complement encoding
- Given an unsigned integer A which is represented in the binary system using n bits, then:
  - The 1’s complement encoding of A = $(2^n - 1) - A$
  - The 2’s complement encoding of A = $2^n – A$
- E.g.:
  - given A = 0110
  - The 1’s complement encoding of A = $(2^4 - 1) - 0110 = 1001$
  - The 2’s complement encoding of A = $2^4 - 0110 = 1010$

Representation of signed integer

- A signed integer number –A is represented by the 2’s complement encoding of A
- E.g.: Represent signed number A = -70 using 8 bits

A = 70 = 0100 0110

```
1’s comp.:        1011 1001
                 +        1
2’s comp.:        1011 1010
```

Hence: -70 => 1011 1010

Representation of Real number

- General rule:
  - In order to represent real number in computer, we use Floating point number

Representation of Real number

- A floating point number has the form of:

  $$X = M * R^E$$

  in which:
  - M is mantissa
  - R is radix (base), usually be 2 or 10.
  - E is exponent
- Given R, to represent a real number X, we need to determine M and E (as integer number)

Representation of Real number

- E.g.: Using R = 10, given two real numbers N1 and N2 represented in floating point with:
  - M1 = -15 và E1 = +12
  - M2 = +314 và E2 = -2

Determine the values of N1 and N2?

Operations on floating point real number

- Given two floating point real numbers:
  - $N1 = M1 \times R^{E1}$ and $N2 = M2 \times R^{E2}$
- Add/Subtract:

$$N1 \pm N2 = (M1 \times R^{E1-E2} \pm M2) \times R^{E2} ,$$

- Multiple and divide

$$N1 \times N2 = (M1 \times M2) \times R^{E1+E2}$$

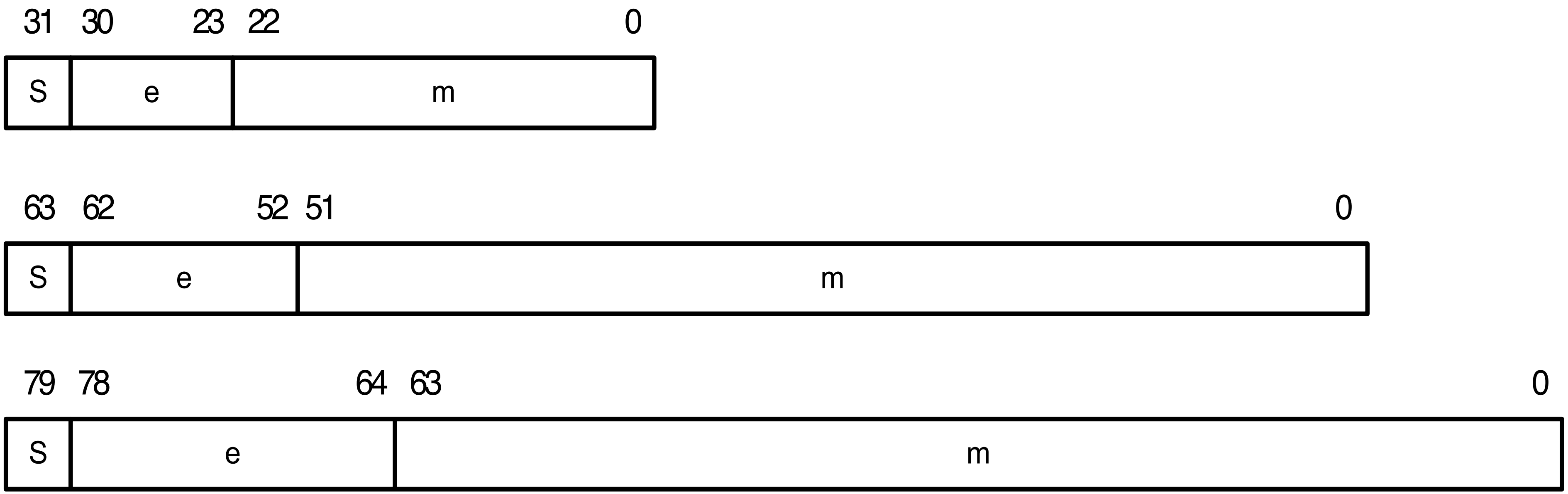
$$N1 / N2 = (M1 / M2) / R^{E1-E2}$$

IEEE 754/85 format

- Is used to represent Floating point number
- Radix R = 2
- Consist of 3 types:
  - Single precision, using 32-bit
  - Double precision, using 64-bit
  - Extended double precision, using 80-bit

IEEE 754/85 format

Encoding format



IEEE 754/85 format

- S is signed bit: S=0 for positive and and S=1 for negative number.
- e is excess of exponent E: E = e – b

  where b is bias:
  - With 32-bit : b = 127, then E = e - 127
  - With 64-bit : b = 1023, then E = e - 1023
  - With 80-bit : b = 16383, then E = e – 16383
- m is fraction part of M: M = 1.m

IEEE 754/85 format

S	e	m

- Value of X is determined as follow:

$$X = (-1)^{S} \times 1.m \times 2^{e-b}$$

Example

- E.g.1: Given X which is represented in IEEE 754 32 bit:

1100 0001 0101 0110 0000 0000 0000 0000

Determine value of X.

- E.g.2: Given Y which is represented in IEEE 754 32 bit:

0011 1111 1000 0000 0000 0000 0000 0000

Determine value of Y.

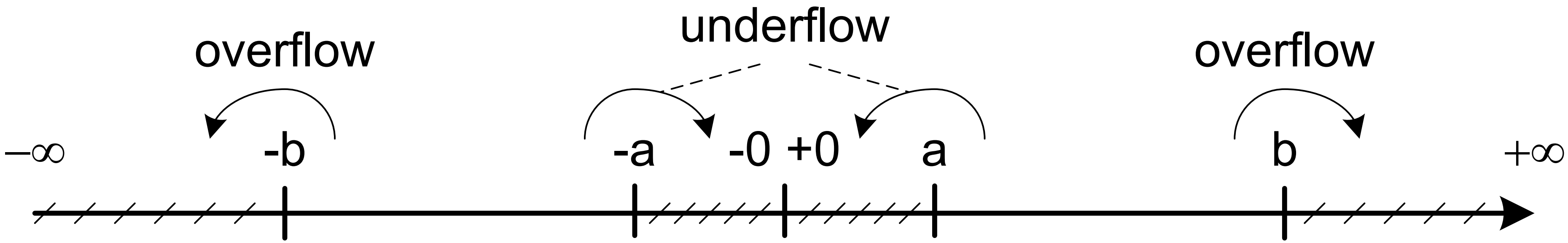
Example

- E.g. 3: Given a real number X = 6.375. Represent X in IEEE 754 using 32 bit

Note

- If all bits of e are 0 and all bits of m are 0, then X = ± 0
- If all bits of e are 1 and all bits of m are 0, then X = ± ∞
- If all bits of e are 1 and at least one bit of m is 1, the X is not a number (NaN)

Representation range



- 32 bit: $a = 2^{-127} \approx 10^{-38}$ $b = 2^{+127} \approx 10^{+38}$
- 64 bit: $a = 2^{-1023} \approx 10^{-308}$ $b = 2^{+1023} \approx 10^{+308}$
- 80 bit: $a = 2^{-16383} \approx 10^{-4932}$ $b = 2^{+16383} \approx 10^{+4932}$

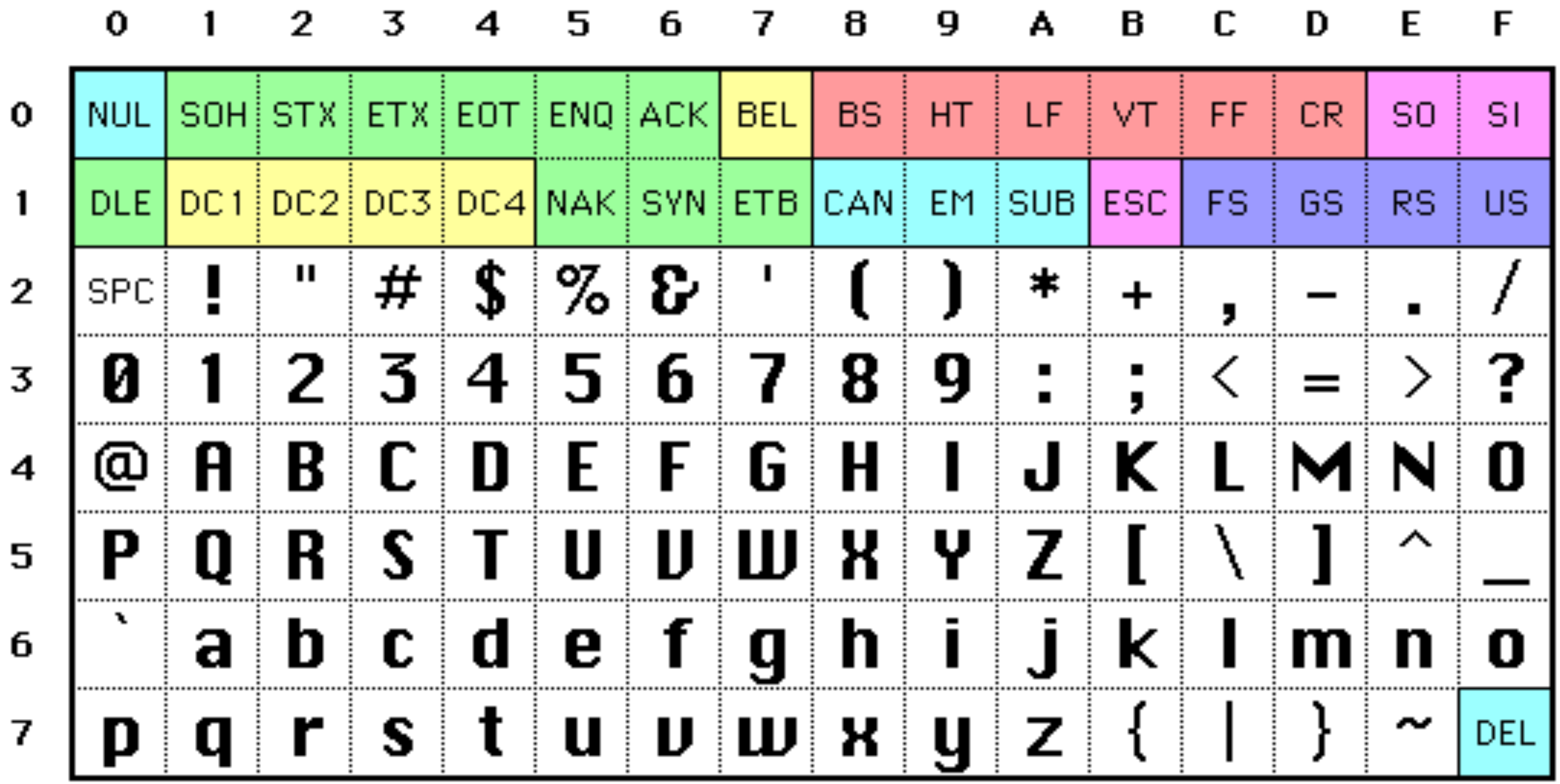
Representation of characters

- **General rule:**
  - Convert characters to binary.
  - The number of bits used to encode each character depend on encoding set.
  - E.g. :
    - ASCII uses 8 bits
    - Unicode uses 16 bits.

ASCII

- Proposed by ANSI (American National Standard Institute)
- ASCII uses 8 bit, hence it can represent 256 characters
- Character has code in range: $00_{16} \div FF_{16}$

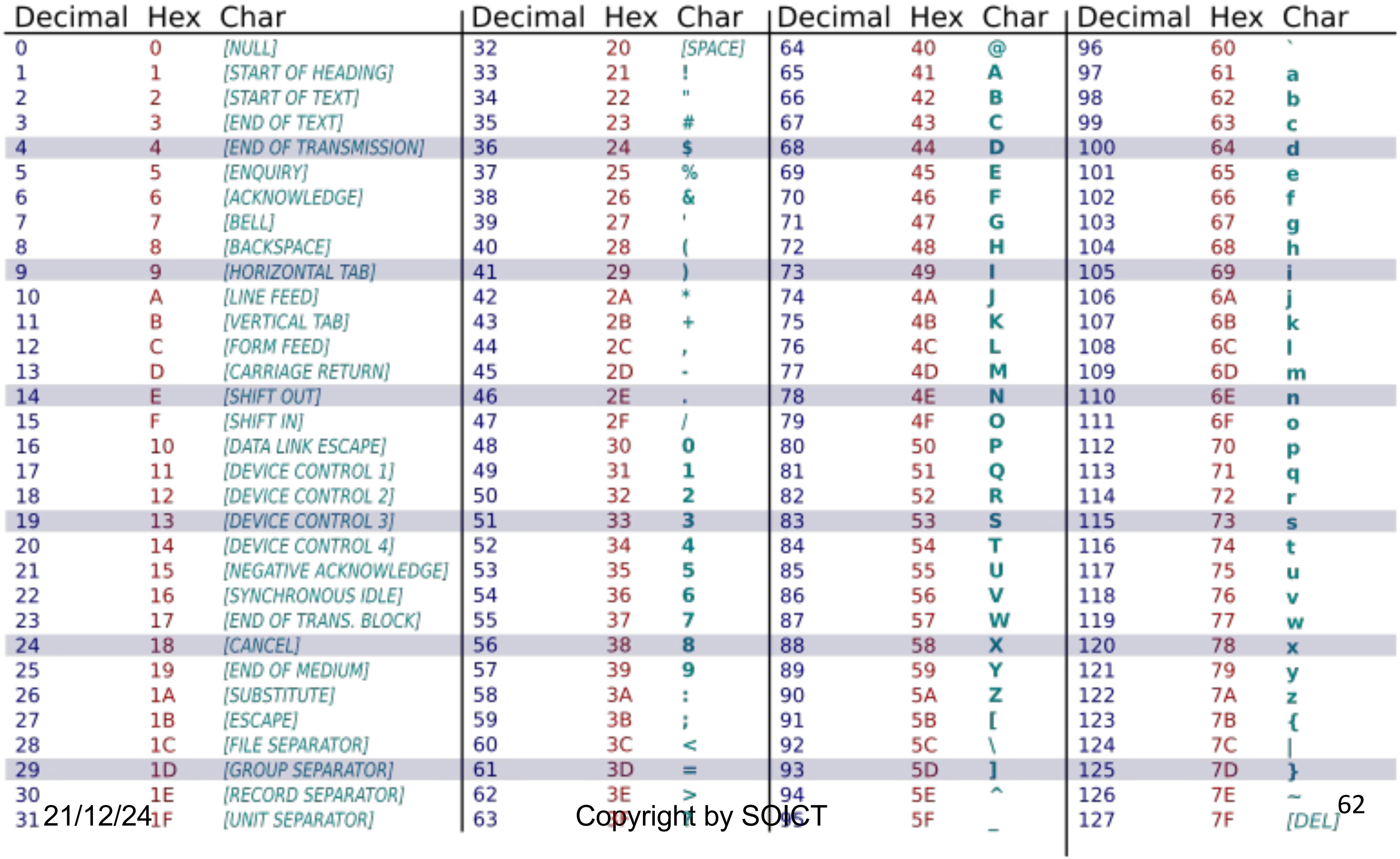
ASCII Chart



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

🟥 **Physical Device Controls: Format Effectors**

ASCII TABLE



Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

ASCII

- 95 displayable characters has code:20_{16} ÷ $7E_{16,}$ in which:
  - 'A' ÷ 'Z' has code: 41_{16} ÷ $5A_{16}$
  - 'a' ÷ 'z' has code: 61_{16} ÷ $7A_{16}$
  - '0' ÷ '9' has code: 30_{16} ÷ 39_{16}