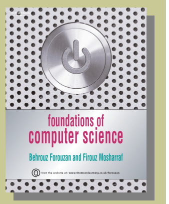


Number Systems

Instructor: Tsung-Che Chiang
tcchiang@ieee.org
Department of Computer Science and Information Engineering
National Taiwan Normal University

Outline

- Positional Number Systems
- Conversion between Positional Number Systems
- Non-positional Number Systems
- Summary

Introduction

- A number system defines how a number can be represented using distinct symbols.
- A number can be represented differently in different systems.
  - For example, the two numbers $(2A)_{16}$ and $(52)_8$ both refer to the same quantity, $(42)_{10}$.

Positional Number Systems

- In a positional number system, the position a symbol occupies in the number determines the value it represents.
- In this system, a number represented as:

$$\pm (S_{k-1} \ldots S_2\, S_1\, S_0 .\, S_{-1}\, S_{-2} \ldots S_{-l})_b$$

has the value of:

$$n = \pm\ S_{k-1} \times b^{k-1} + \ldots + S_1 \times b^1 + S_0 \times b^0 \ +\ S_{-1} \times b^{-1} + S_{-2} \times b^{-2} + \ldots + S_{-l} \times b^{-l}$$

b is the base.

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

The Decimal System (base 10)

- The word decimal is derived from the Latin root decem (ten).
- Base $b = 10$.
- Ten symbols: $S = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- The symbols in this system are often referred to as decimal digits or just digits.

The Decimal System (base 10)

- Integer values

	10^{k-1}	10^{k-2}	$\cdots$	10^2	10^1	10^0	Place values
$\pm$	S_{k-1}	S_{k-2}	$\cdots$	S_2	S_1	S_0	Number

$$N = \pm S_{k-1} \times 10^{k-1} + S_{k-2} \times 10^{k-2} + \cdots + S_2 \times 10^2 + S_1 \times 10^1 + S_0 \times 10^0 \quad \text{Values}$$

	10^2	10^1	10^0	Place values
	2	2	4	Number
$N = +$	2×10^2	$+\ 2 \times 10^1$	$+\ 4 \times 10^0$	Values

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

The Decimal System (base 10)

- Real values

Integral part / Fractional part

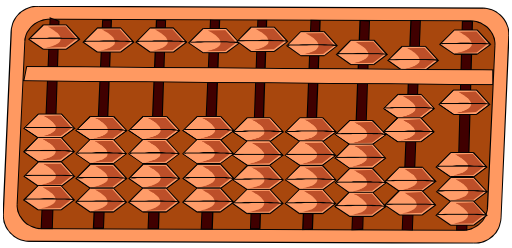
$$R = \pm\; S_{k-1} \times 10^{k-1} + \ldots + S_1 \times 10^1 + S_0 \times 10^0 \;+\; S_{-1} \times 10^{-1} + \ldots + S_{-l} \times 10^{-l}$$

	10^1	10^0		10^{-1}	10^{-2}	Place values
	2	4	•	1	3	Number
R = +	2 × 10	+ 4 × 1	+	1 × 0.1	+ 3 × 0.01	Values

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

The Decimal System (base 10)

- Abacus – a device that uses positional notation to represent a decimal number.

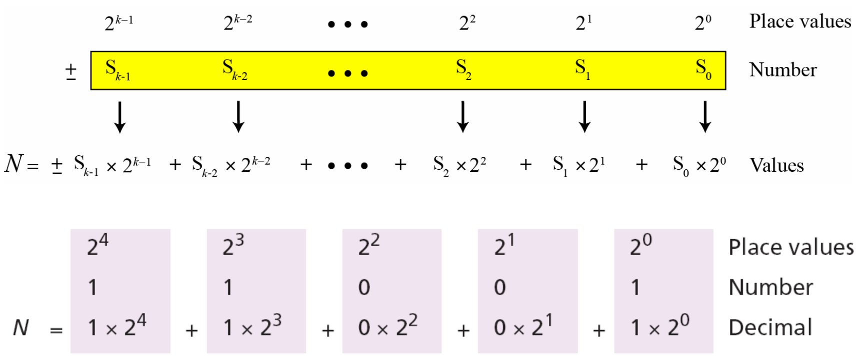


The Binary System (base 2)

- The word binary is derived from the Latin root bini (double).
- Base $b = 2$.
- Ten symbols: $S = \{0, 1\}$
- The symbols in this system are often referred to as binary digits or just bits.

The Binary System (base 2)

- Integer values



$(11001)_2 =$?

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

The Binary System (base 2)

- Real values

	Integral part	.	Fractional part
$R = \pm$	$S_{k-1} \times 2^{k-1} + \ldots + S_1 \times 2^1 + S_0 \times 2^0$	+	$S_{-1} \times 2^{-1} + \ldots + S_{-l} \times 2^{-l}$

		2^2		2^1		2^0		2^{-1}		2^{-2}		Place values
		1		0		1	•	1		1		Number
R	=	1×2^2	+	0×2^1	+	1×2^0	+	1×2^{-1}	+	1×2^{-2}		Values

$(101.11)_2 =$?

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

The Binary System (base 2)

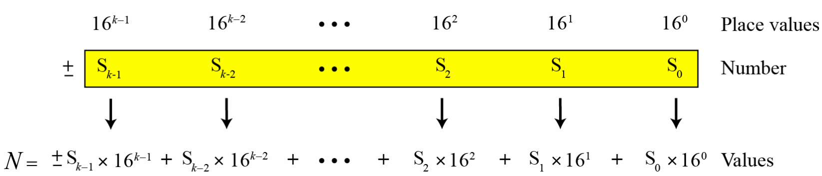
- Chinese binary system

The Hexadecimal System (base 16)

- The word hexadecimal is derived from the Greek root hex (six) and Latin root decem (ten).
- Base $b = 16$.
- Ten symbols: $S = \{0, 1, \ldots, 8, 9, A, B, C, D, E, F\}$
- The symbols in this system are often referred to as hexadecimal digits.

The Hexadecimal System (base 16)

- Integer values



		16^2		16^1		16^0	Place values
		2		A		E	Number
N	=	2×16^2	+	10×16^1	+	14×16^0	Values

$(2AE)_{16} =$ **?**

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

The Octal System (base 8)

- The word octal is derived from the Latin root octo (eight).
- Base $b = 8$.
- Ten symbols: $S = \{0, 1, 2, 3, 4, 5, 6, 7\}$

The Octal System (base 8)

Place values	8^{k-1}	8^{k-2}	$\cdots$	8^2	8^1	8^0
Number ($\pm$)	S_{k-1}	S_{k-2}	$\cdots$	S_2	S_1	S_0

Values:

$$N = \pm S_{k-1} \times 8^{k-1} + S_{k-2} \times 8^{k-2} + \cdots + S_2 \times 8^2 + S_1 \times 8^1 + S_0 \times 8^0$$

	8^3	8^2	8^1	8^0	Place values
	1	2	5	6	Number
N =	1×8^3 +	2×8^2 +	5×8^1 +	6×8^0	Values

$(1256)_8 =$?

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Summary of the Four Positional Number Systems

Table 2.1 Summary of the four positional number systems

System	*Base*	*Symbols*	*Examples*
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	2345.56
Binary	2	0, 1	$(1001.11)_2$
Octal	8	0, 1, 2, 3, 4, 5, 6, 7	$(156.23)_8$
Hexadecimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	$(A2C.A1)_{16}$

Summary of the Four Positional Number Systems

Table 2.2 Comparison of numbers in the four systems

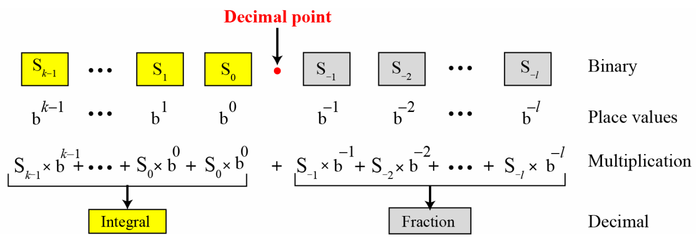
Decimal	*Binary*	*Octal*	*Hexadecimal*
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Conversion between Number Systems

- We will introduce how to do the following conversions:
  - Binary/Hex/Octal → Decimal.
  - Decimal → Binary/Hex/Octal.
  - Binary ↔ Hex/Octal

Conversion between Number Systems

- Binary/Hex/Octal → Decimal.



Binary	1		1		0	•	1		1
Place values	2^2		2^1		2^0		2^{-1}		2^{-2}
Partial results	4	+	2	+	0	+	0.5	+	0.25

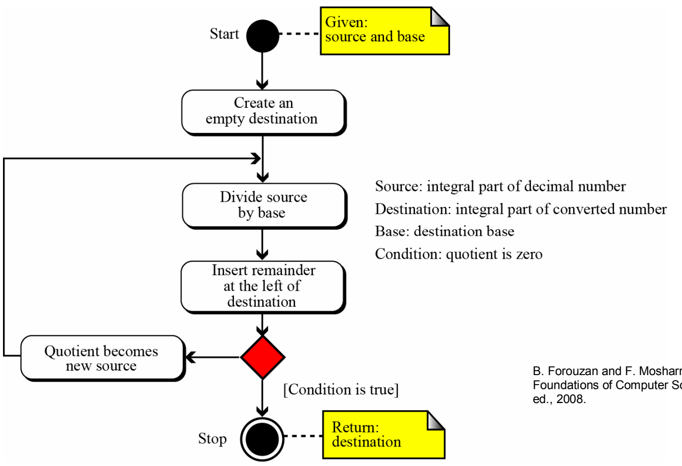
Decimal: 6.75

$(110.11)_2 = (6.75)_{10}$

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Decimal-Others Conversion

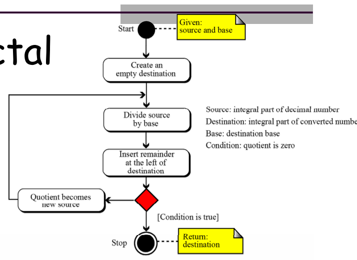
- Decimal → Binary/Hex/Octal (Integral part)

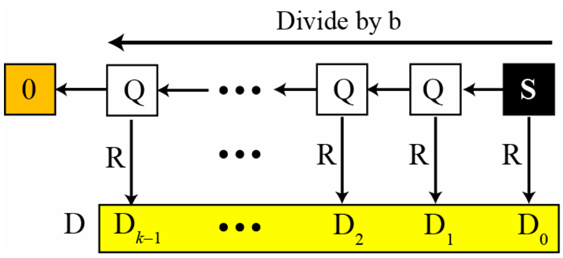


B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Decimal-Others Conversion

- Decimal → Binary/Hex/Octal





Q: Quotients
R: Remainders
S: Source
D: Destination
D_i: Destination digit

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

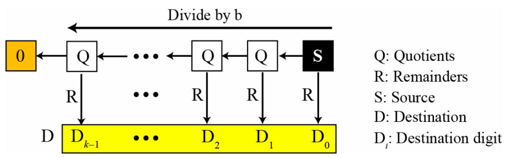
Decimal-Others Conversion

- Decimal
  → Binary/Hex/Octal

$35 = 17 \cdot 2 + 1 \quad = \quad 17 \cdot 2^1 + 1 \cdot 2^0$

$= (8 \cdot 2 + 1) \cdot 2 + 1 \quad = 8 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$

…



$(35)_{10} = (100011)_2$

0	←	1	←	2	←	4	←	8	←	17	←	**35**	Decimal
		↓		↓		↓		↓		↓		↓	
		1		0		0		0		1		1	Binary

Decimal-Others Conversion

- Decimal
  → Binary/Hex/Octal

0	←	7	←	**126**	Decimal
		↓		↓	
		7		E	Hexadecimal

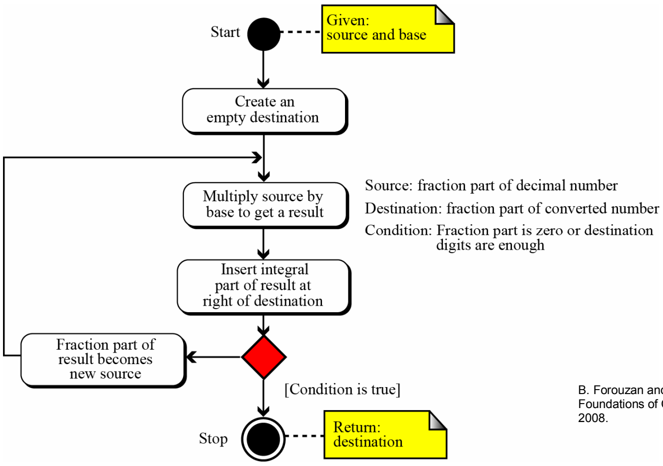
$(126)_{10} =$?

0	←	1	←	15	←	**126**	Decimal
		↓		↓		↓	
		1		7		6	Octal

$(126)_{10} =$?

Decimal-Others Conversion

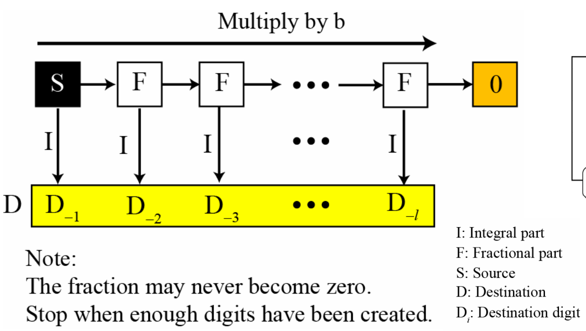
- Decimal → Binary/Hex/Octal (Fractional part)



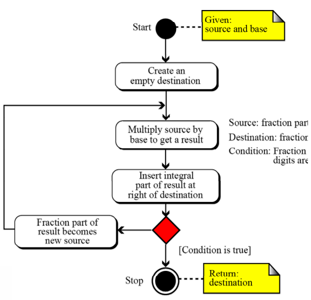
B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Decimal-Others Conversion

- Decimal → Binary/Hex/Octal (Fractional part)



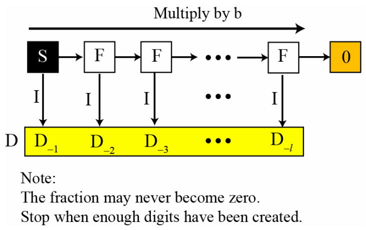
Note:
The fraction may never become zero.
Stop when enough digits have been created.



B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Decimal-Others Conversion

- Decimal → Binary/Hex/Octal (Fractional part)



$$
\begin{aligned}
0.625 &= 2^{-1}\cdot(2^{1}\cdot 0.625) = 2^{-1}\cdot(1.25)\\
&= 2^{-1}\cdot(1 + 0.25)\\
&= 2^{-1}\cdot(1 + 2^{-1}\cdot(2^{1}\cdot 0.25))\\
&= 2^{-1}\cdot(1 + 2^{-1}\cdot(0 + 0.5))\\
&= 2^{-1}\cdot(1 + 2^{-1}\cdot(0 + 2^{-1}\cdot(2^{1}\cdot 0.5)))\\
&= 2^{-1}\cdot(1 + 2^{-1}\cdot(0 + 2^{-1}\cdot(1 + 0)))
\end{aligned}
$$

Decimal		0.625 →	0.25 →	0.50 →	0.00	
		↓	↓	↓		
Binary	•	1	0	1		$(0.625)_{10} = (0.101)_2$

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Decimal-Others Conversion

- Decimal → Binary/Hex/Octal (Fractional part)

Decimal	0 ←	11 ←	178		0.6 →	0.6	
		↓	↓		↓		
Hexadecimal		B	2	•	9		$(178.6)_{10} = (\ ?\ )_{16}$

Decimal		0.634 →	0.072 →	0.576 →	0.608 →	0.864	
		↓	↓	↓	↓		
Octal	•	5	0	4	4		$(0.634)_{10} = (\ ?\ )_8$ (maximum of four digits)

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Decimal-Others Conversion

- Decimal → Binary/Hex/Octal
  - An alternative method: (usu. for small values)

    To break the number as the sum of numbers that are equivalent to the binary place values shown.

Place values	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal equivalent	128	64	32	16	8	4	2	1

Decimal 165 =	128	+	0	+	32	+	0	+	0	+	4	+	0	+	1
Binary	1		0		1		0		0		1		0		1

Decimal-Others Conversion

- Decimal → Binary/Hex/Octal
  - An alternative method: (usu. for small values)

Place values	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}
Decimal equivalent	1/2	1/4	1/8	1/16	1/32	1/64	1/128

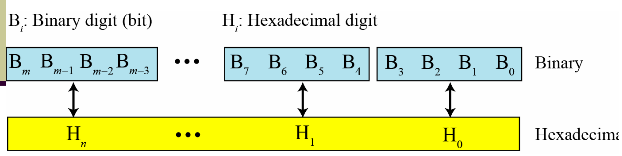
Decimal = 27/64	16/64	+	8/64	+	2/64	+	1/64
	1/4	+	1/8	+	1/32	+	1/64

Decimal 27/64 =	0	+	1/4	+	1/8	+	0	+	1/32	+	1/64
Binary	0		1		1		0		1		1

Binary-Hexadecimal Conversion

- four-bit grouping

$(10011100010)_2 \rightarrow (?)_{16}$



Decimal	Binary	Hexadecimal
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	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

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Binary-Hexadecimal Conversion

- Binary → Hex

$(10011100010)_2 \rightarrow (100\ 1110\ 0010)_2 \rightarrow (4E2)_{16}$

- Hex → Binary

$(24C)_{16} \rightarrow (0010\ 0100\ 1100)_2 \rightarrow (1001001100)_2$

Binary-Hexadecimal Conversion

- Binary → Hex

$(11010101011)_2 \rightarrow$ **?**

- Hex → Binary

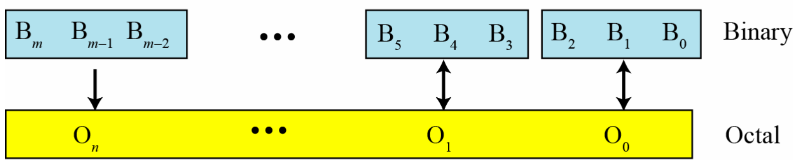
$(58F)_{16} \rightarrow$ **?**

Binary-Octal Conversion

- three-bit grouping

Decimal	Binary	Octal
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7

B_i: Binary digit (bit) O_i: Octal digit



B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Binary-Octal Conversion

- Binary → Octal

$(101110010)_2 \rightarrow (101\ 110\ 010)_2 \rightarrow (562)_8$

- Octal → Binary

$(24)_8 \rightarrow (010\ 100)_2 \rightarrow (10100)_2$

Binary-Octal Conversion

- Binary → Octal

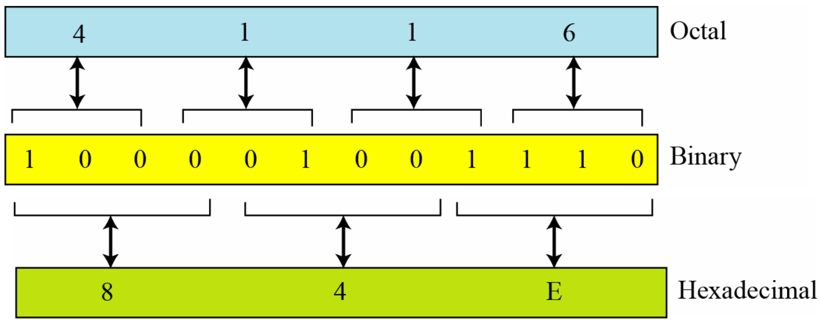
$(11100110011)_2 \rightarrow$?

- Octal → Binary

$(765)_8 \rightarrow$?

Octal-Hex Conversion

- Convert with the aid of binary systems



B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Number of Digits

- Quiz:

  Find the minimum number of binary digits required to store decimal integers with a maximum of six digits.

Number of Digits

- How can we know the number of digits required to store a k-digit-base-b_1 integral value in the base-b_2 system?
  - Maximum k-digit value in base b_1: $(b_1^k - 1)$
    Maximum x-digit value in base b_2: $(b_2^x - 1)$

$$(b_2^x - 1) \geq (b_1^k - 1) \quad \Rightarrow x \geq k \cdot (\log b_1 / \log b_2)$$

Number of Digits

- Quiz:

  Find the minimum number of binary digits required to store decimal integers with a maximum of six digits.

$k = 6, b_1 = 10, b_2 = 2.$

$x \geq 6 \cdot (\log b_1 / \log b_2) = 6 \cdot (1/0.30103) = 19.9$

$\Rightarrow x = 20$

$2^{19} = 524288$
$2^{20} = 1048576$

Non-positional Number System

- A non-positional number system still uses a limited number of symbols in which each symbol has a value.
- However, the position a symbol occupies in the number normally bears no relation to its value—the value of each symbol is fixed.
- To find the value of a number, we add the value of all symbols present in the representation.

Non-positional Number System

- In this system, a number is represented as:

$$S_{k-1} \ldots S_2\, S_1\, S_0 \bullet S_{-1}\, S_{-2} \ldots S_{-l}$$

and has the value of:

$$n = \pm \underbrace{S_{k-1} + \ldots + S_1 + S_0}_{\text{Integral part}} + \underbrace{S_{-1} + S_{-2} + \ldots + S_{-l}}_{\text{Fractional part}}$$

There are some exceptions to the addition rule we just mentioned, as shown in the following example.

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Non-positional Number System

- Example: Roman numerals

Table 2.3 Values of symbols in the Roman number system

Symbol	*I*	*V*	*X*	*L*	*C*	*D*	*M*
Value	1	5	10	50	100	500	1000

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Non-positional Number System

- Example: Roman numerals
  - When a symbol with a smaller value is placed **after** a symbol having an equal or larger value, the values are added.

III	→	1 + 1 + 1	=	3
VIII	→	5 + 1 + 1 + 1	=	8
XVIII	→	10 + 5 + 1 + 1 + 1	=	18
LXXII	→	50 + 10 + 10 + 1 + 1	=	72
CI	→	100 + 1	=	101
MMVII	→	1000 + 1000 + 5 + 1 + 1	=	2007
MDC	→	1000 + 500 + 100	=	1600

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Table 2.3 Values of symbols in the Roman number system

Symbol	*I*	*V*	*X*	*L*	*C*	*D*	*M*
Value	1	5	10	50	100	500	1000

Non-positional Number System

- Example: Roman numerals
  - When a symbol with a smaller value is placed **before** a symbol having a larger value, the smaller value is subtracted from the larger one.

IV	→	5 − 1	=	4
XIX	→	10 + (10 −1)	=	19

Table 2.3 Values of symbols in the Roman number system

Symbol	*I*	*V*	*X*	*L*	*C*	*D*	*M*
Value	1	5	10	50	100	500	1000

B. Forouzan and F. Mosharraf, Foundations of Computer Science, 2nd ed., 2008.

Non-positional Number System

- Example: Roman numerals
  - For other rules, please refer to the text book.

Summary

- Number systems
  - Positional vs. Non-positional
- Positional systems
  - Decimal
  - Binary
  - Octal
  - Hexadecimal

  } *conversion*
- Non-positional systems
  - Roman numeral