Number Systems

Lecture 2 – Monday February 27, 2017

Outline

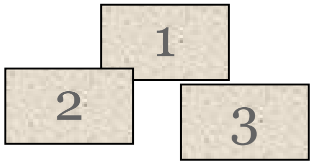
- Data Representation
- Number Systems
- Decimal System
- Binary System
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- Octal System
- Hexadecimal or “Hex” System
- Converting from Base to Base
- Summary

Outline

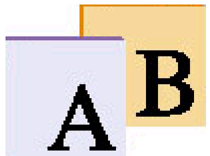
- **<u>Data Representation</u>**
- Number Systems
- Decimal System
- Binary System
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- Octal System
- Hexadecimal or “Hex” System
- Converting from Base to Base
- Summary

Data Representation

- Information comes in many forms

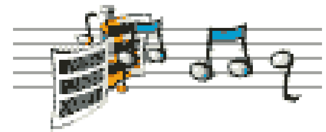


Numbers



Text

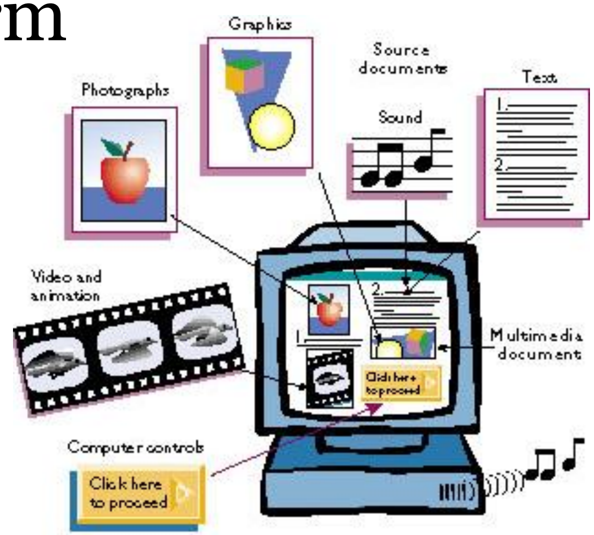
Images



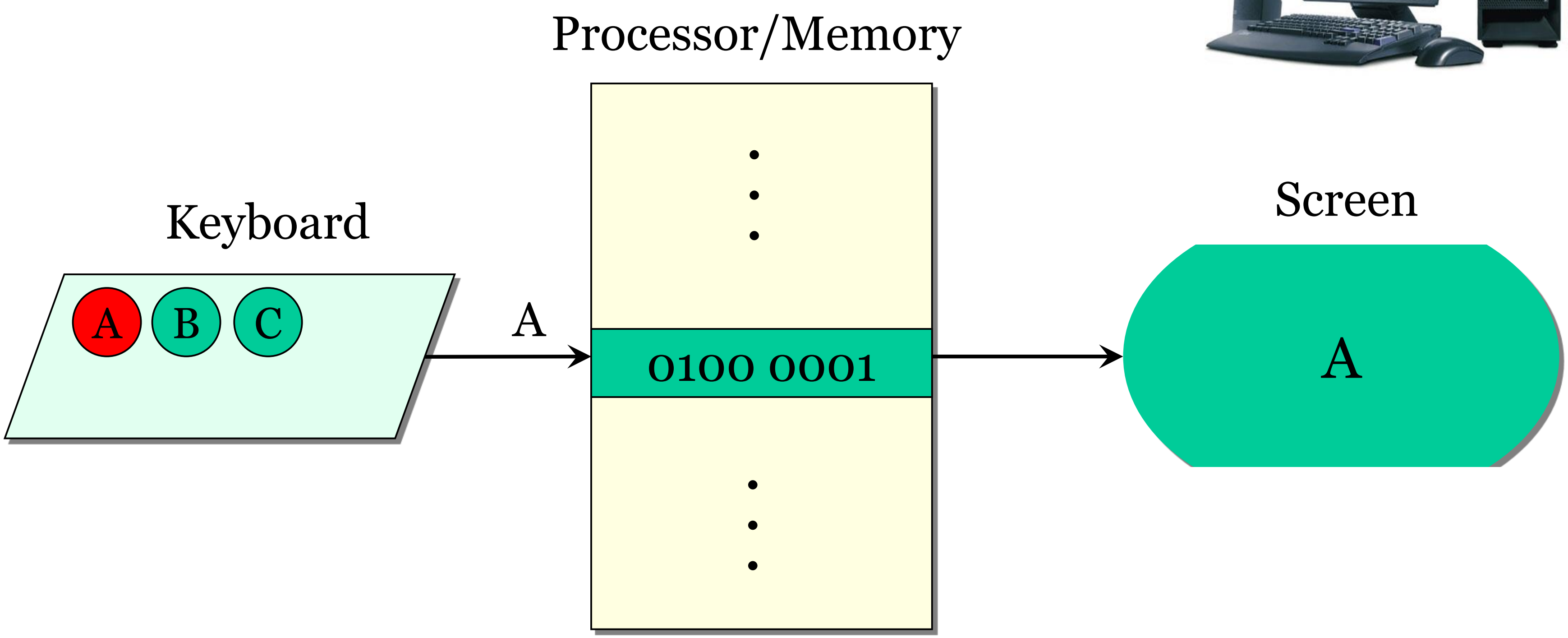
Audio

Video

- Computers process and store information in digital form



Data Representation



External Representation

Internal Representation

External Representation

Outline

- Data Representation
- **Number Systems**
- Decimal System
- Binary System
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- Octal System
- Hexadecimal or “Hex” System
- Converting from Base to Base
- Summary

Number Systems

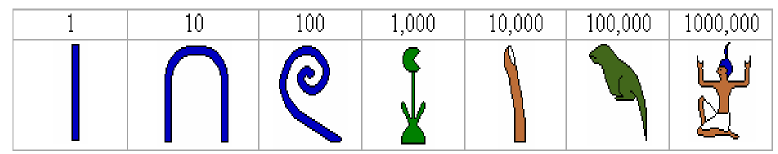
Number Systems

Positional Systems

Decimal	Binary	Octal	Hexadecimal
Base 10	Base 2	Base 8	Base 16
0.....9	0,1	0...7	0...9, A...F
	157=10011101	157=235	157=9D

Non-Positional Systems

Egyptian numerals



157= 

Roman numerals

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

157=CLVII

More info: http://gwydir.demon.co.uk/jo/numbers/index.htm

Number Systems

Number Systems

Positional Systems

1 ⇓ Hundreds

5 ⇓ Tens

7 ⇓ Units

$$1 * 10^2 + 5 * 10^1 + 7 * 10^0$$

In positional systems, each position is related to the next by a constant multiplier, a common ratio, called the base of that numeral system.

Non-Positional Systems

C ⇓ L ⇓ VII ⇓

There is no any relation between the positions. There is no any common base.

Outline

- Data Representation
- Number Systems
- **<u>Decimal System</u>**
- Binary System
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- Octal System
- Hexadecimal or “Hex” System
- Converting from Base to Base
- Summary

Decimal System

- A number is represented as a string of digits.
- Each digit position has a *weight* associated with it.
- Number's value = a *weighted sum* of the digits ⇒ $D = \sum_{i=-n}^{p-1} d_i 10^i$
- Fractions: Weights that are negative powers of 10

Example: 425.97

Digits	4	2	5	.	9	7
Weights	10^2	10^1	10^0	.	10^{-1}	10^{-2}
$425.97 = 4 * 10^2 + 2 * 10^1 + 5 * 10^0 + 9 * 10^{-1} + 7 * 10^{-2}$						

$$425.97 = \sum_{i=-2}^{2} d_i 10^i$$

Outline

- Data Representation
- Number Systems
- Decimal System
- **Binary System**
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- Octal System
- Hexadecimal or “Hex” System
- Converting from Base to Base
- Summary

Binary System

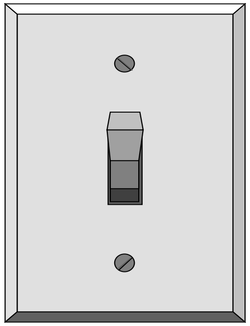
- The "base" is 2 instead of 10
- Meaning: the weights are powers of 2 instead of powers of 10. Digits are called "bits," for "**bi**nary digi**ts**."

Example: $19_{10} \Rightarrow 10011_2$

Digits	1	0	0	1	1
Weights	2^4	2^3	2^2	2^1	2^0
$1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0 = 19$					

Binary System

- **Bit Basics**



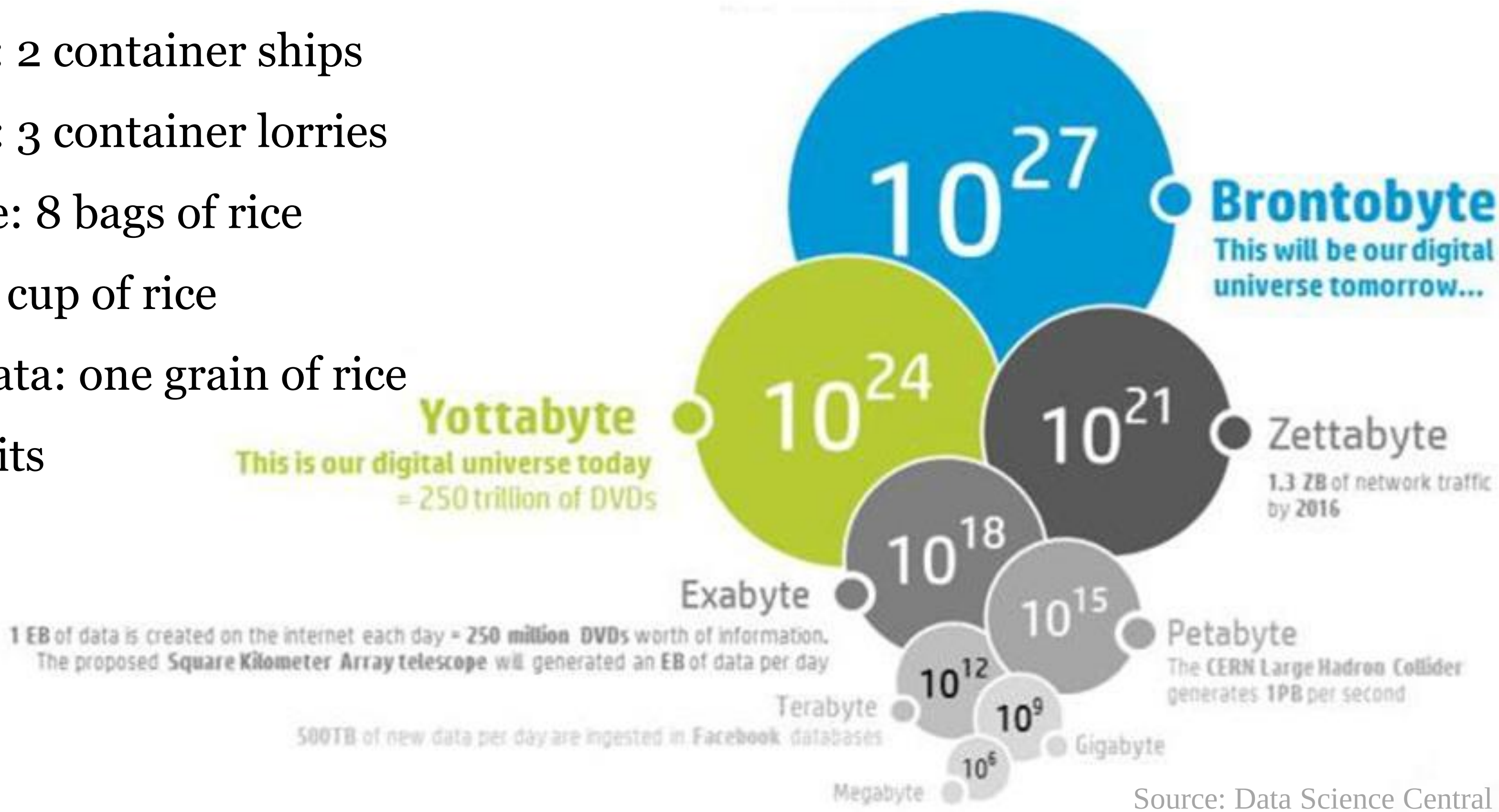
On

A bit (binary digit)

- is the smallest unit of information
- can have two values: 1 or 0
- can represent numbers, codes, or instructions
- High=ON=1 and Low=OFF=0

Binary System

- **Bit Basics**
  - Zettabyte: fills the Pacific ocean
  - Exabyte: covers Germany twice
  - Petabyte: covers Manhattan
  - Terabyte: 2 container ships
  - Gigabyte: 3 container lorries
  - Megabyte: 8 bags of rice
  - Kilobyte: cup of rice
  - Byte of data: one grain of rice
  - Byte=8 bits



Source: Data Science Central

Binary System

- **Bits as numbers**
  - A keyboard translates the touch on the numeric keypad into series of 0s and 1s
  - Each number then is looked at as a component of its positional values (each a power of 2).

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
…	….

Binary System

- **Bits as codes**

  - *ASCII* (American Standard Code for Information Interchange): It is Most widely used code, represents each character as a unique 8-bit code

  - *Unicode:* A coding scheme that supports 65,000 unique characters.

Character	ASCII binary code
A	01000001
B	01000010
C	01000011
D	01000100
E	01000101
F	01000110
G	01000111
H	01001000
I	01001001
J	01001010
K	01001011
L	01001100
M	01001101
N	01001110
O	01001111
P	01010000
Q	01010001
R	01010010
S	01010011
T	01010100
U	01010101
V	01010110
W	01010111
X	01011000
Y	01011001
Z	01011010
0	00110000
1	00110001
2	00110010
3	00110011
4	00110100
5	00110101
6	00110110
7	00110111
8	00111000
9	00111001

Outline

- Data Representation
- Number Systems
- Decimal System
- Binary System
- **Binary to Decimal Conversion**
- Decimal to Binary Conversion
- Octal System
- Hexadecimal or “Hex” System
- Converting from Base to Base
- Summary

Binary to Decimal Conversion

- Write down the binary number
- Write down the weight (power of 2) corresponding to each position in the binary number
- Multiply each digit by its weight
- Add all products.

Binary to Decimal Conversion

Example: Convert the binary number $(1101101)_2$ to a decimal number.

Digits	1	1	0	1	1	0	1
Weights	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Weight Value	64	32	16	8	4	2	1
Digits x Weight Value	1x64	1x32	0x16	1x8	1x4	0x2	1x1
Decimal	64 + 32 + 0 + 8 + 4 + 0+ 1 = 109						

$(1101101)_2 = 109_{10}$

Binary to Decimal Conversion

Exercise

Convert the following binary numbers into decimal:

111011011

1011011.0110

00110.11001

Outline

- Data Representation
- Number Systems
- Decimal System
- Binary System
- Binary to Decimal Conversion
- **Decimal to Binary Conversion**
- Octal System
- Hexadecimal or “Hex” System
- Converting from Base to Base
- Summary

Decimal to Binary Conversion

- **Integer Part**
  - Successively divide number by 2, taking remainder as result.
  - Repeat division and writing down the remainder until the quotient equals 0
  - Write the binary number by writing the remainders from bottom to top.

Decimal to Binary Conversion

- **Integer Part**

Example-1: Convert 57_{10} to binary.

Division	Quotient	Remainder
57/2	28	1
28/2	14	0
14/2	7	0
7/2	3	1
3/2	1	1
1/2	0	1

(LSB: Least Significant Bit)

(MSB: Most Significant Bit)

$57_{10} = 111001_2$

Decimal to Binary Conversion

- **Integer Part**

Example-2: Convert 26_{10} to binary.

Division	Quotient	Remainder
26/2	13	0
13/2	6	1
6/2	3	0
3/2	1	1
1/2	0	1

(LSB: Least Significant Bit)

$26_{10} = 11010_2$

(MSB: Most Significant Bit)

Decimal to Binary Conversion

- **Integer Part**

Exercise

Convert the following decimal numbers into binary:

19, 123, 65, 78 and 111

Decimal to Binary Conversion

- **Fraction Part**
  - Successively multiply number by 2, taking integer part as result and chopping off integer part before next iteration.
  - Read the binary number by reading the integer parts from top to bottom.
  - May be unending!

Decimal to Binary Conversion

- **Fraction Part**

Example: convert .3 to binary.

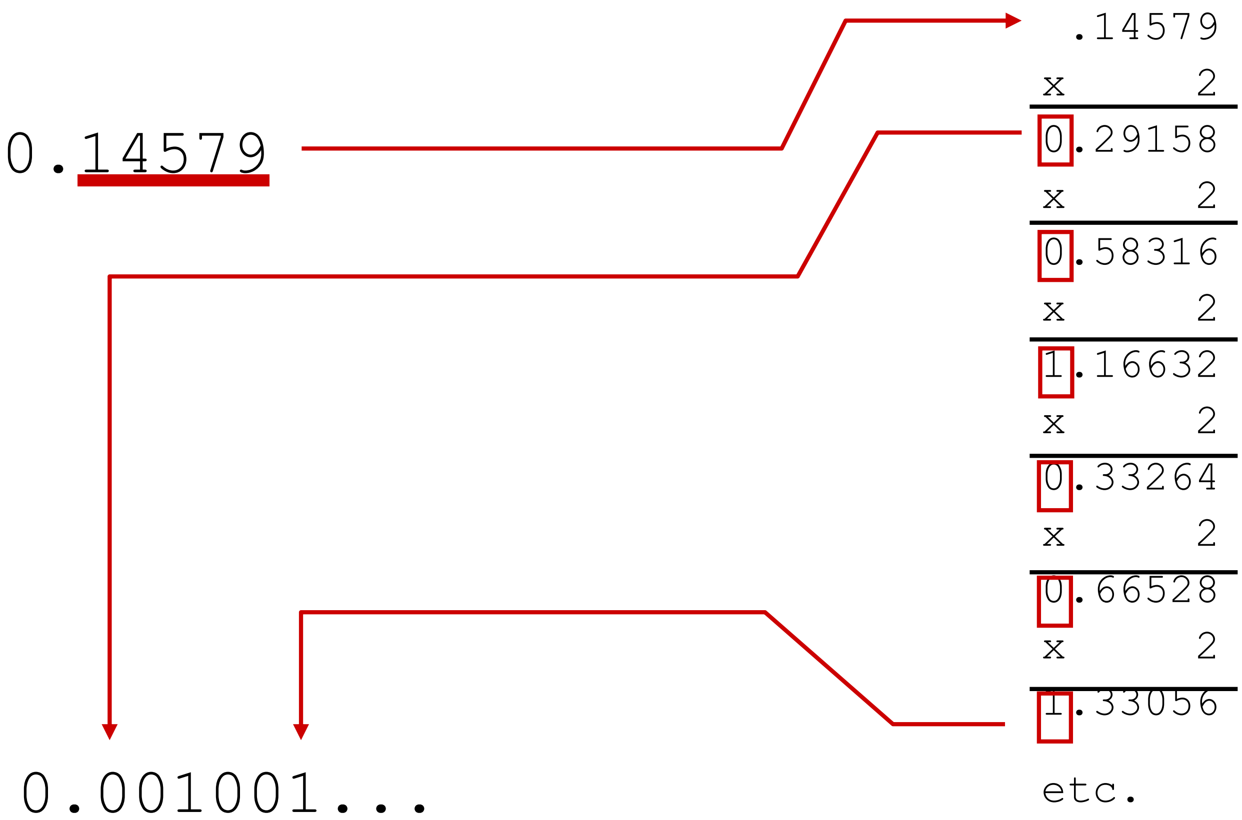
Multiplication	Quotient	Integer Part
.3 * 2	.6	0
.6 * 2	1.2	1
.2 * 2	.4	0
.4 * 2	.8	0
.8 * 2	1.6	1
.6 * 2	1.2	1, etc.

$.3_{10} = .010011_2$

Decimal to Binary Conversion

- **Fraction Part**

Example: convert 0.14579 to binary.



Decimal to Binary Conversion

- **Fraction Part**

Exercise

Convert the following decimal numbers into binary:

0.6, 75.325 and 320.7

Outline

- Data Representation
- Number Systems
- Decimal System
- Binary System
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- **Octal System**
- Hexadecimal or "Hex" System
- Converting from Base to Base
- Summary

Octal System

Octal and hexadecimal number systems are great ways to concisely represent a **bit pattern**.

Each **octal digit** is exactly equivalent to **3 bits**, and

Each **hexadecimal digit** is exactly equivalent to **4 bits**.

As such, a large bytestring, like can be concisely represented in octal as or in hex as simply by observation.

100100101011010010101010 → 44532252

Binary → Octal

100100101011010010101010 → 92b45a

Binary → Hexadecimal

Octal System

- The "base" is 8 instead of 10

Decimal	0	1	2	3	4	5	6	7	8	9
Octal	0	1	2	3	4	5	6	7	10	11

- **Why Octal?**
  - Memory addresses
  - Unix file system permissions have three sets (user, group, others)
  - In avionics, ARINC 429 word labels are almost always expressed in octal
  - ...

Outline

- Data Representation
- Number Systems
- Decimal System
- Binary System
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- Octal System
- **Hexadecimal or “Hex” System**
- Converting from Base to Base
- Summary

Hexadecimal or "Hex" System

- The "base" is 16 instead of 10

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

- **Why Hexadecimal?**

Scientists working on the early computers figured they'd go blind trying to read through strings of zeros and ones.

So, when writing down numbers, they converted the zeros and ones to hexadecimal. It was an easy way to illustrate the zeros and ones. (Every hex digit would represent 4 binary digits.)

$$1101\ 0100\ 1011\ 0110_2 = D4B6_{16}$$

Think of hexadecimal as a short handed form for binary.

Outline

- Data Representation
- Number Systems
- Decimal System
- Binary System
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- Octal System
- Hexadecimal or “Hex” System
- **<u>Converting from Base to Base</u>**
- Summary

Converting from Base to Base

Decimal	Binary	Octal	Hex
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

Converting from Base to Base

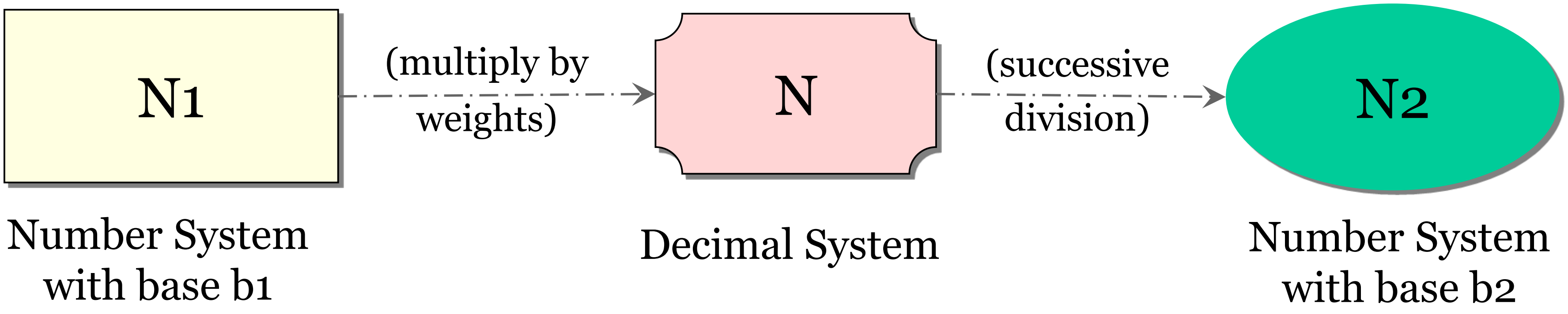
Given: a number N1 in base b1.

Required: convert N1 to a number N2 in base b2

Solution:

1. Convert N1 to the number N in base 10

2. Convert N to the number N2 in base b2



Converting from Base to Base

Example: Convert the number 1315_6 to a number in base 11.

Conversion from base 6 to base 10

Digits	1	3	1	5
Weights	6^3	6^2	6^1	6^0
Weight Value	216	36	6	1
Digits x Weight Value	1x216	3x36	1x6	5x1
Decimal	216+108+6+5=335			

Conversion from base 10 to base 11

Division	Quotient	Remainder
335/11	30	5
30/11	2	8
2/11	0	2

$1315_6 = 285_{11}$

Converting from Base to Base

Special Case-1: Converting from Binary to Octal

- Every octal digit can be converted to exactly three binary digits (8 is 2^3). Partition the binary number into **groups of three bits**, starting with the rightmost bit.
- Then replace each three-bit group by the corresponding octal digit.

Example: Convert 001011110_2 to octal.

$$001\ 011\ 110_2 = 136_8$$

$$\Downarrow \quad \Downarrow \quad \Downarrow$$

$$1 \quad 3 \quad 6$$

Converting from Base to Base

Special Case-2: Converting from Binary to Hex

- Since 16 is 2^4, every hexadecimal digit can be converted to exactly four binary digits. Partition the binary number into **groups of four bits**, starting with the rightmost bit.
- Then replace each four-bit group by the corresponding hex digit.

Example: Convert 1101010010110110_2 to hex.

$$1101\ 0100\ 1011\ 0110_2 = D4B6_{16}$$

1101	0100	1011	0110
⇓	⇓	⇓	⇓
D	4	B	6

Outline

- Data Representation
- Number Systems
- Decimal System
- Binary System
- Binary to Decimal Conversion
- Decimal to Binary Conversion
- Octal System
- Hexadecimal or “Hex” System
- Converting from Base to Base
- **Summary**

Summary

- The external representation of information is the way information is represented by humans and the way it is entered at a keyboard or displayed on a printer or screen.
- The internal representation is the way it is stored in the memory of a computer.
- In positional systems, each position is related to the next by a constant multiplier, a common ratio, called the base of that numeral system.
- Binary is a base-2 positional numbering system.
- Hex and octal are useful for representing multibit binary numbers because their bases are integer multiples of 2.
- Regardless of using hex or base10 system, hardware wise, a computer only handles binary numbers.