جامعة الملك سعود
King Saud University

CSC 220: Computer Organization

Unit 1
Number Systems

Prepared by:

Md Saiful Islam, PhD

Updated by:

Isra Al-Turaiki, PhD

Department of Computer Science

College of Computer and Information Sciences

Aug 2023

Overview

- Common Number Systems
- Conversion Among Bases
- Binary Coded Decimal (BCD)

Chapter-1

M. Morris Mano, Charles R. Kime and Tom Martin, **Logic and Computer Design Fundamentals**, Global (5^{th}) Edition, Pearson Education Limited, 2016. ISBN: 9781292096124

Number Systems – Representation

A number with *radix* (or base) r

1. contains r digits: 0, 1, 2, . . ., r-1
2. is represented by a string of digits:

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

   in which $0 \le A_i < r$ and . is the *radix point*.
3. The string of digits represents the power series:

digit position base

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

(Integer Portio (Fraction Portion)

Number Systems – Examples

		General	Decimal	Binary
Radix (Base)		r	10	2
Digits		0 => r - 1	0 => 9	0 => 1
	0	r^0	1	1
	1	r^1	10	2
	2	r^2	100	4
	3	r^3	1000	8
Powers of	4	r^4	10,000	16
Radix	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

Commonly Occurring Bases

Name	Radix	Digits
Binary	2	0,1
Octal	8	0,1,2,3,4,5,6,7
Decimal	10	0,1,2,3,4,5,6,7,8,9
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Common Number Systems

TABLE 1-3
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

System	Used by humans?	Used in computers?
Decimal	Yes	No
Binary	No	Yes
Octal	No	No
Hexadecimal	No	No

Positional number systems: decimal

- Numbers consist of a bunch of digits, each with a weight:

A_i	1	6	2	.	3	7	5	Digits
r^i	100	10	1		1/10	1/100	1/1000	Weights

- The weights are all powers of the base, which is 10. We can rewrite the weights like this:

1	6	2	.	3	7	5	Digits
10^2	10^1	10^0		10^{-1}	10^{-2}	10^{-3}	Weights

- To find the decimal value of a number, multiply each digit by its weight and sum the products.

$$(1 \times 10^2) + (6 \times 10^1) + (2 \times 10^0) + (3 \times 10^{-1}) + (7 \times 10^{-2}) + (5 \times 10^{-3}) = 162.375$$

↓

Weighted Multiplication

Positional number systems: binary

- We can use the same trick for binary, or base 2, numbers. The only difference is that the weights are powers of 2.
- For example, here is 1101.01 in binary:

1	1	0	1	.	0	1	Binary digits, or bits
2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	Weights (in base 2)

- The decimal value is:

$$(1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}) =$$
$$8 + 4 + 0 + 1 + 0 + 0.25 = 13.25$$

Powers of 2:

$2^0 = 1$	$2^4 = 16$	$2^8 = 256$
$2^1 = 2$	$2^5 = 32$	$2^9 = 512$
$2^2 = 4$	$2^6 = 64$	$2^{10} = 1024$
$2^3 = 8$	$2^7 = 128$	

Special Powers of 2

2^{10} (1024) is Kilo
2^{20} (1,048,576) is Mega
2^{30} (1,073, 741,824)is Giga

Base 8 - Octal

- The octal system uses 8 digits:

  0 1 2 3 4 5 6 7

- Earlier in this history of computing, octal was used as a shorthand for binary numbers. (Now hexadecimal is more common.)
- Since $8 = 2^3$, one octal digit is equivalent to 3 binary digits.
  - Numbers like 67 are easier to work with than 110111.
- Still shows up in some places. For instance, file access permissions in Unix file systems.

Decimal	Binary	Octal
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7

Base 16 is useful too

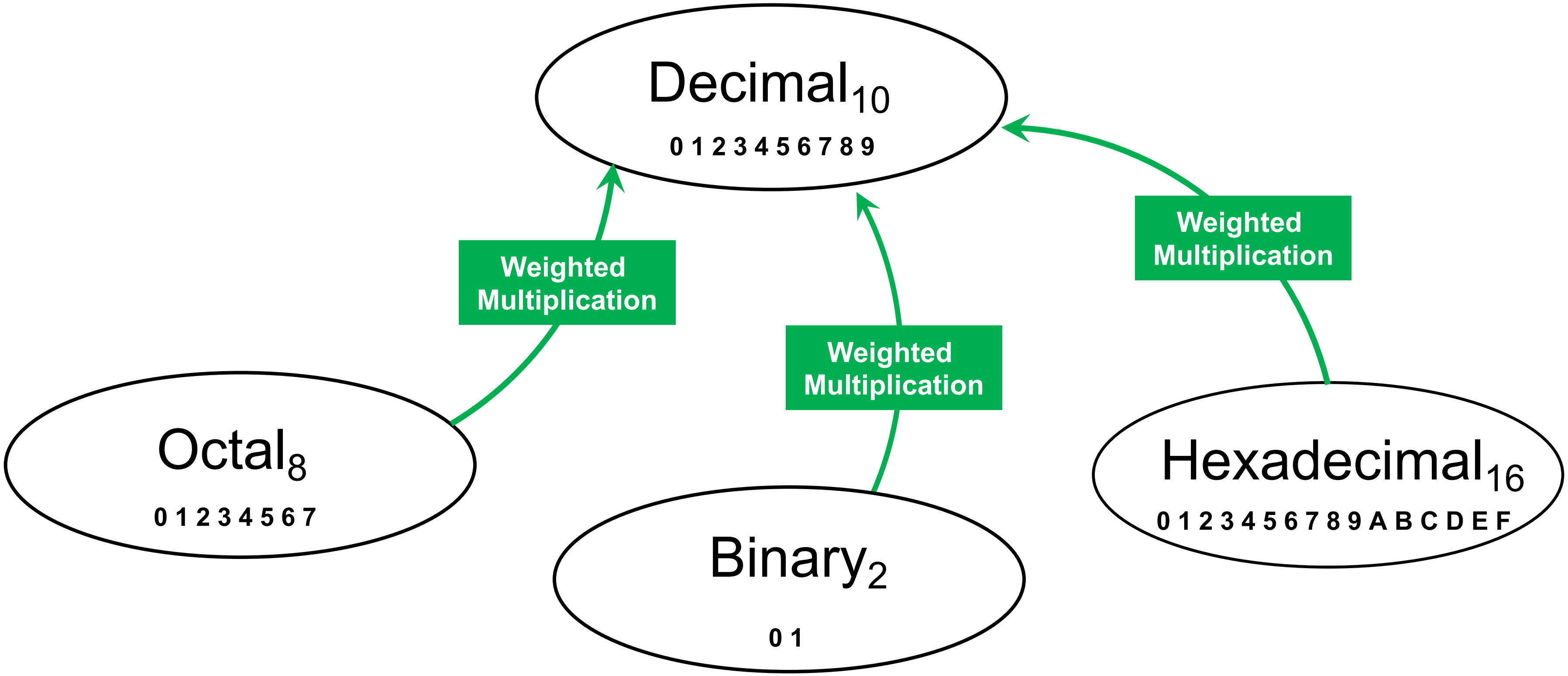
- The hexadecimal system uses 16 digits:

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

- For our purposes, base 16 is most useful as a "shorthand" notation for binary numbers.
  - Since $16 = 2^4$, one hexadecimal digit is equivalent to 4 binary digits.
  - It's often easier to work with a number like B4 instead of 10110100.
- Hex is frequently used to specify things like:
  - IP addresses such as: 80.AE.05.27
  - RGB colors: C0C0FF

Decimal	Binary	Hex
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

Conversion Among Bases



Converting to Decimal- 1

Binary to Decimal

- Technique: **Weighted Multiplication**
  - **Multiply** each bit by 2^n, where 2^n is the "weight" of the bit
  - n is the position of the bit, starting from 0 on the right
  - Add the results

- Example:

101011_2 =>

digit		power of 2		
1	x	2^0	=	1
1	x	2^1	=	2
0	x	2^2	=	0
1	x	2^3	=	8
0	x	2^4	=	0
1	x	2^5	=	32
				43_{10}

Converting to Decimal- 2

Binary Fractions to Decimal

- Technique: **Weighted Multiplication**

- Example:

10.1011 =>

	digit		power of 2		
	1	x	2^{-4}	=	0.0625
	1	x	2^{-3}	=	0.125
	0	x	2^{-2}	=	0.0
	1	x	2^{-1}	=	0.5
	0	x	2^{0}	=	0.0
	1	x	2^{1}	=	2.0
					2.6875

Converting to Decimal-3

Octal to Decimal

- Technique: **Weighted Multiplication**
  - **Multiply** each bit by 8^n, where 8^n is the "weight" of the bit
  - n is the position of the bit, starting from 0 on the right
  - Add the results

- Example:

$$
\begin{aligned}
724_8 \Rightarrow \quad & 4 \times 8^0 = 4 \\
& 2 \times 8^1 = 16 \\
& 7 \times 8^2 = 448 \\
& \overline{468_{10}}
\end{aligned}
$$

Converting to Decimal-4

Hexadecimal to Decimal

- Technique: **Weighted Multiplication**
  - **Multiply** each bit by 16^n, where 16^n is the "weight" of the bit
  - n is the position of the bit, starting from 0 on the right
  - Add the results

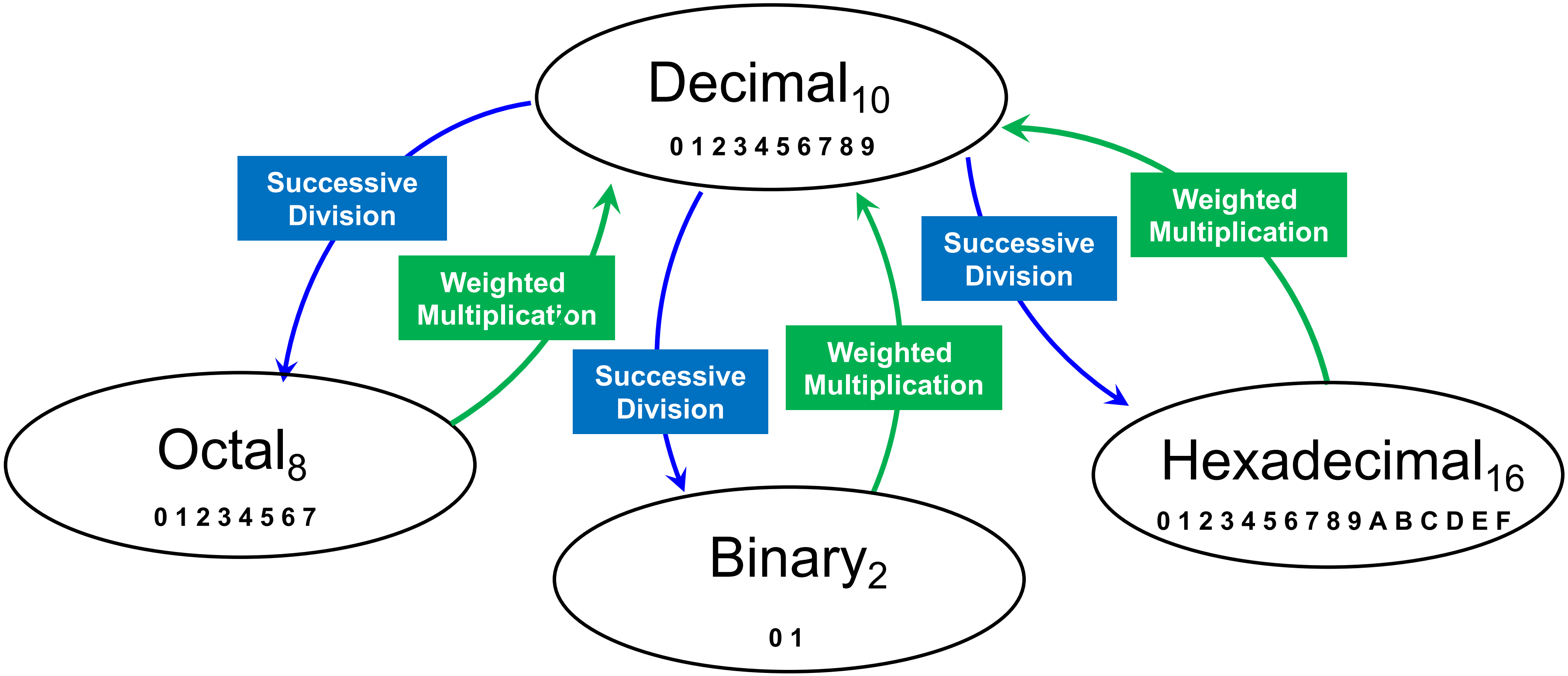
- Example:

$$
\begin{aligned}
ABC_{16} \Rightarrow \quad & C \times 16^0 = 12 \times 1 = 12 \\
& B \times 16^1 = 11 \times 16 = 176 \\
& A \times 16^2 = 10 \times 256 = 2560 \\
& \overline{2748_{10}}
\end{aligned}
$$

Summary: Converting to Decimal

- Technique: Weighted Multiplication
  - **Multiply** each bit by b^n, where b is the "base"
  - n is the position of the bit, starting from 0 on the right
  - Add the results

Conversion Among Bases



Converting from Decimal-1

Decimal to Binary

- Technique: **Successive Division**
  - Divide by two, keep track of the remainder.
  - Repeat until quotient is zero

- **Example:**

$125_{10} = ?_2$

Divisor	Quotient	Remainder	
2	125		
2	62	1	LSB
2	31	0	
2	15	1	
2	7	1	
2	3	1	
2	1	1	
	0	1	MSB

$125_{10} = 1111101_2$

Converting from Decimal-1

Decimal to Octal

- Technique: **Successive Division**
  - Divide by 8 and keep track of the remainder
  - Repeat until quotient is zero

- Example:

$1234_{10} = ?_8$

8	1234		
8	154	2	LSB
8	19	2	
8	2	3	
	0	2	MSB

$1234_{10} = 2322_8$

Converting from Decimal-3

Decimal to Hexadecimal

- Technique: **Successive Division**
  - Divide by 16 and keep track of the remainder.
  - Repeat until quotient is zero

- Example:

$1234_{10} = ?_{16}$

16	1234	
16	77	2
16	4	13 = D
	0	4

$1234_{10} = 4D2_{16}$

Summary: Converting from Decimal

- Technique: **Successive Division**
  - **Divide** by the **base**, keep track of the remainder
  - First remainder is bit 0 (LSB, least-significant bit)
  - Second remainder is bit 1..Etc.
  - Repeat until quotient is zero

Conversion of Decimal Fractions to Binary

Technique:

- multiplication by 2 is used instead of division.
- integers are accumulated instead of remainders.
- repeat until fraction is 0 or sufficient accuracy reached.

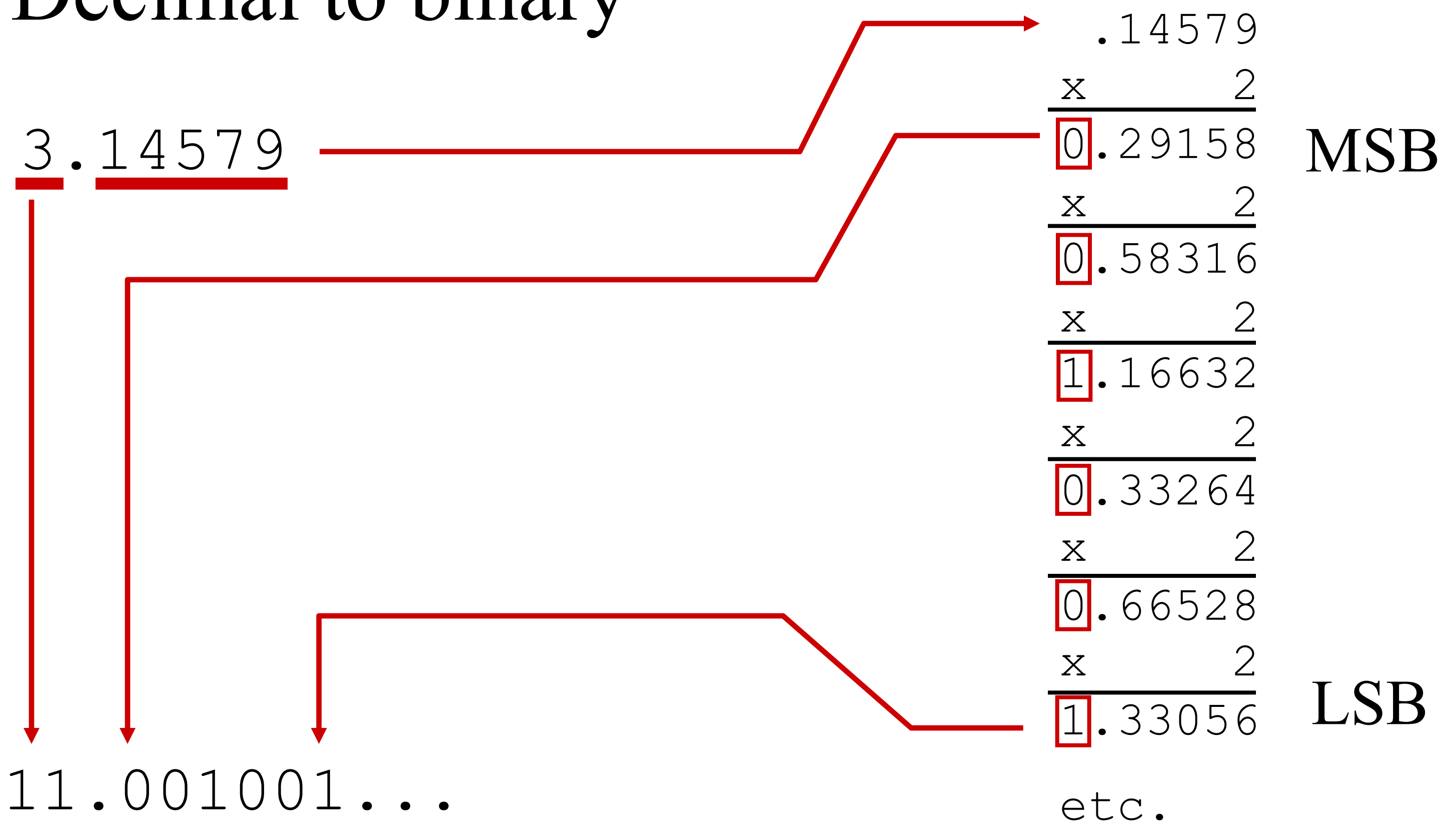
- **Example:**

$0.6875 \times 2 = 1.3750$	Integer $= 1$	↓	Most significant digit
$0.3750 \times 2 = 0.7500$	$= 0$		
$0.7500 \times 2 = 1.5000$	$= 1$		
$0.5000 \times 2 = 1.0000$	$= 1$	↓	Least significant digit

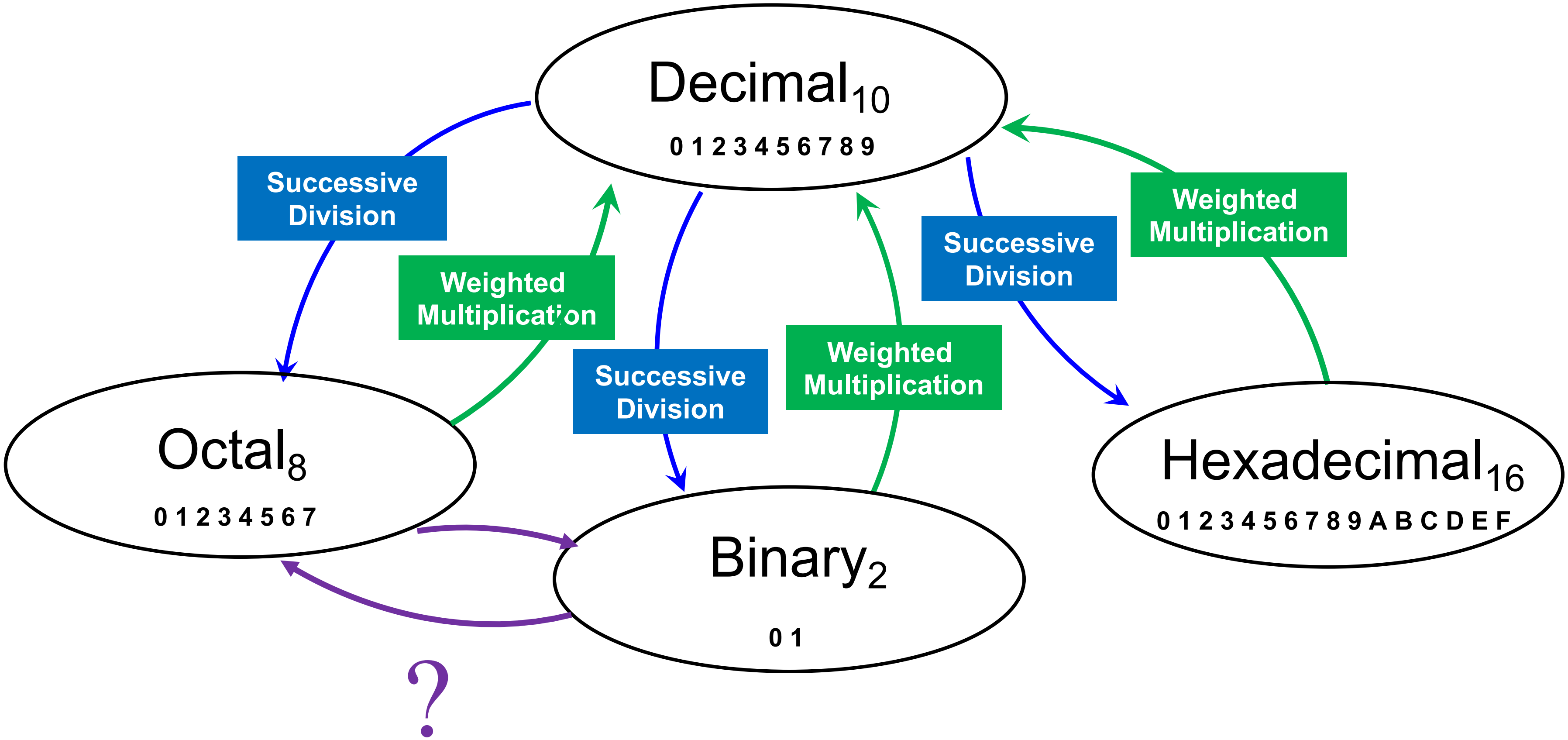
$$(0.6875)_{10} = (0.1011)_2$$

Conversion of Decimal Fractions to Binary

- Decimal to binary



Conversion Among Bases



Converting between Octal and Binary-1

Octal to Binary

- Technique
  - Convert each octal digit to a 3-bit equivalent binary representation

- Example: $705_8 = ?_2$

7 0 5

111 000 101

$705_8 = 111000101_2$

TABLE 1-3
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

Converting between Octal and Binary-2

Binary to Octal

Technique

- Group bits in threes, starting on right
- Convert to octal digits

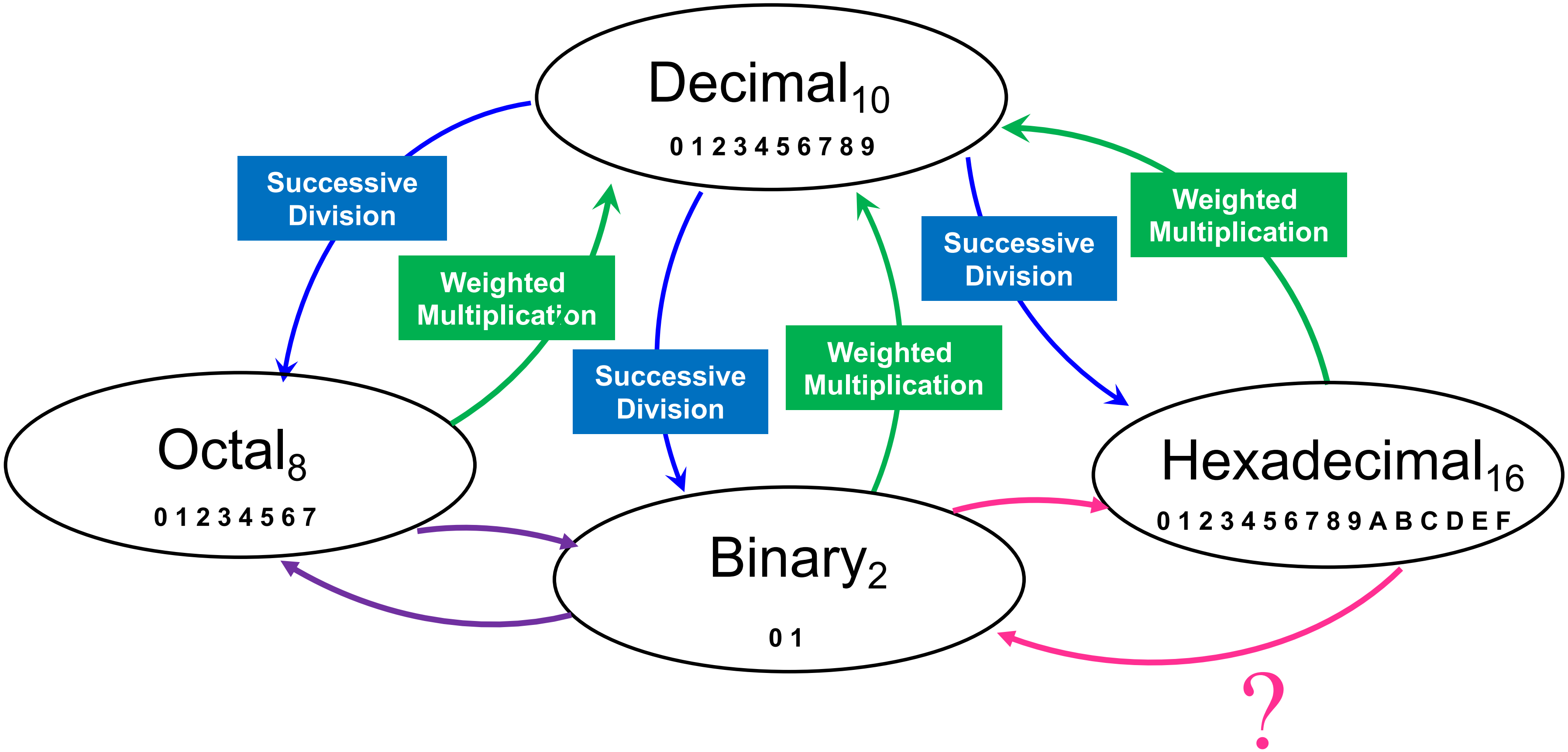
- Example: $1011010111_2 = ?_8$

1 011 010 111

1 3 2 7

$1011010111_2 = 1327_8$

Conversion Among Bases



Converting between Hexadecimal and Binary-1

Hexadecimal to Binary

- Technique
  - Convert each hexadecimal digit to a 4-bit equivalent binary representation

- Example: $10AF_{16} = ?_2$

1 0 A F

↓ ↓ ↓ ↓

0001 0000 1010 1111

$10AF_{16} = 0001000010101111_2$

TABLE 1-3
Numbers with Different Bases

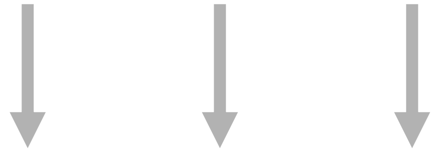
Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Converting between Hexadecimal and Binary-2

Binary to Hexadecimal

- Technique
  - Group bits in fours, starting on right
  - Convert to hexadecimal digits
- Example: $1010111011_2 = ?_{16}$

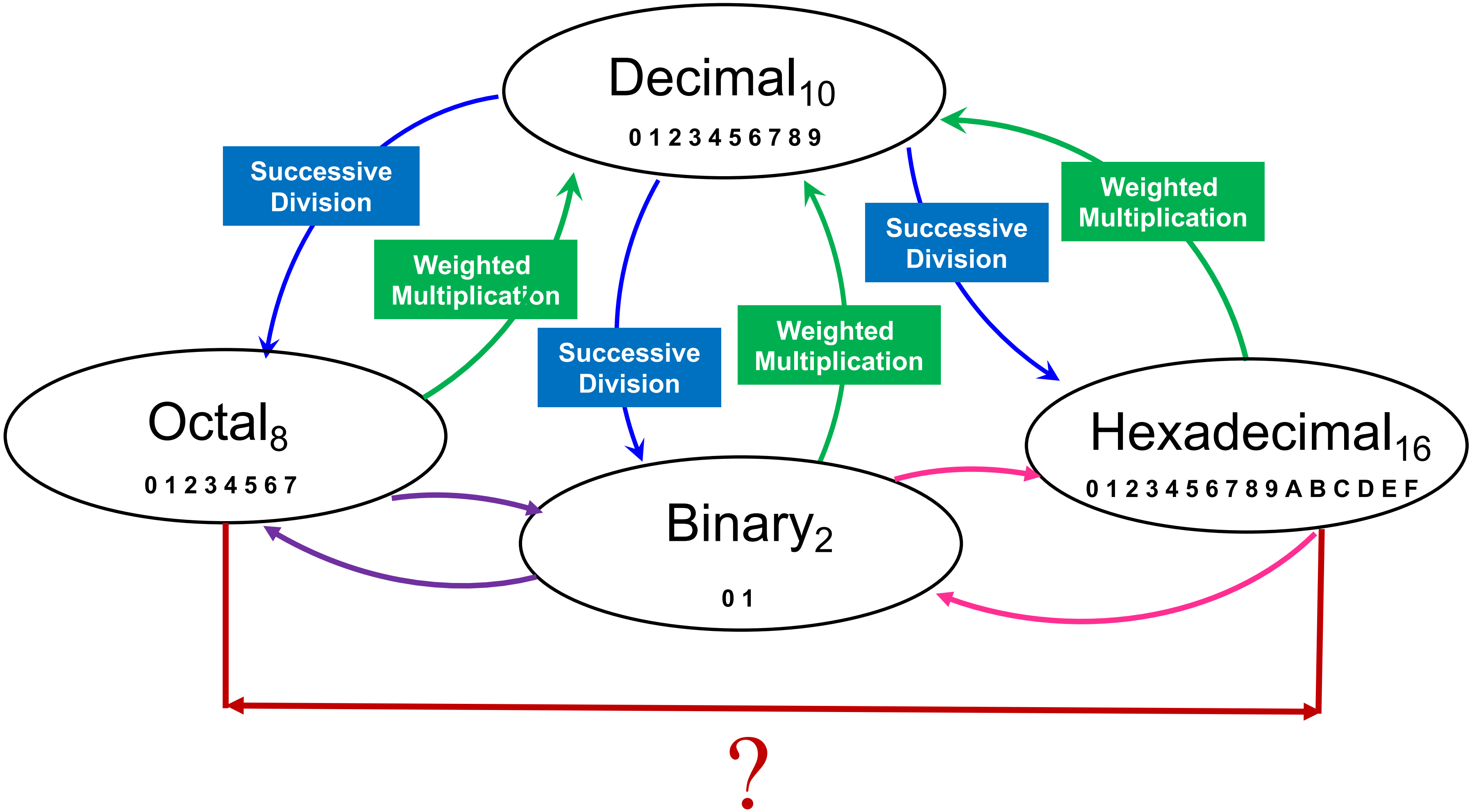
10 1011 1011



2 B B

$1010111011_2 = 2BB_{16}$

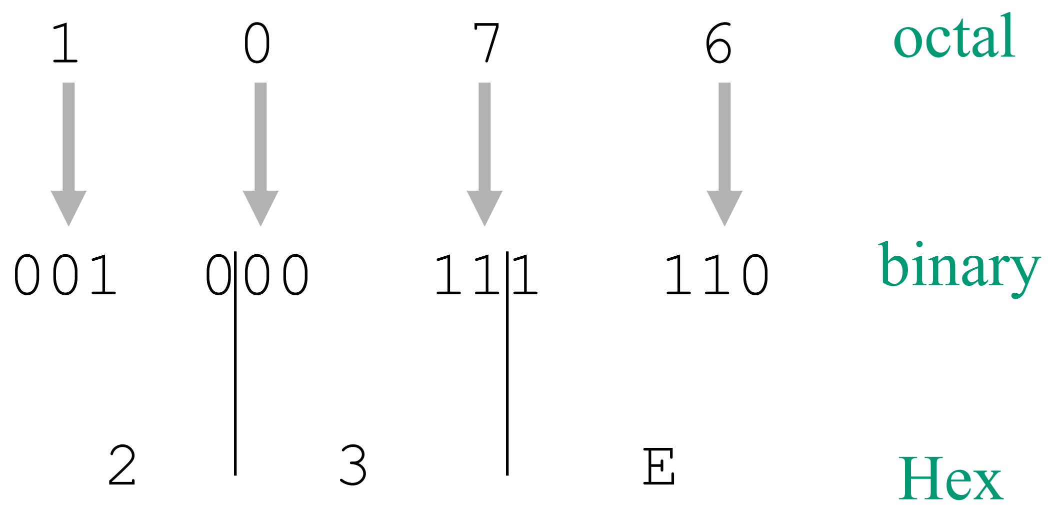
Conversion Among Bases



Converting between Hexadecimal and Octal-1

- Technique
  - Use binary as an intermediary

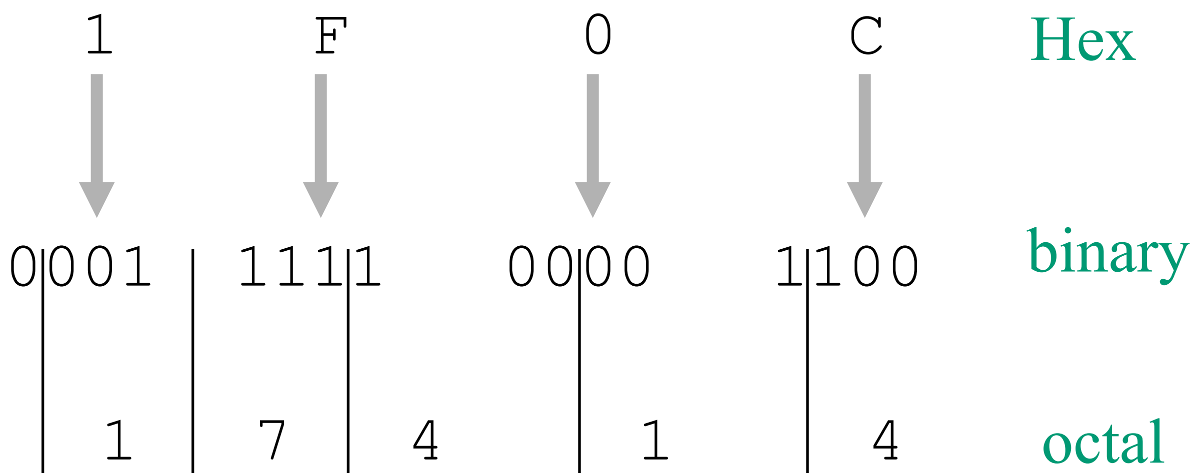
- Example: $1076_8 = ?_{16}$



$1076_8 = 23E_{16}$

Converting between Hexadecimal and Octal-2

- Example: $1F0C_{16} = ?_8$



$$1F0C_{16} = 17414_8$$

4-Bit Binary Coded Decimal (BCD) Systems

- The 4-bit BCD system is usually employed by the computer systems to represent and process numerical data only.
- In the 4-bit BCD system, each digit of the decimal number is encoded to its corresponding 4-bit binary sequence.

Decimal digits	Weighted 4-bit BCD code
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

4-Bit BCD Code

- Represent the decimal number 5327 in BCD code.

4-bit BCD representation of decimal digit 5 is 0101
4-bit BCD representation of decimal digit 3 is 0011
4-bit BCD representation of decimal digit 2 is 0010
4-bit BCD representation of decimal digit 7 is 0111

Therefore, the BCD representation of decimal number 5327 is 0101001100100111.

Warning: Conversion or Coding?

- Do NOT mix up conversion of a decimal number to a binary number with coding a decimal number with a BINARY CODE.

- $13_{10} = 1101_2$ (This is conversion)
- $13 \Leftrightarrow 0001|0011$ (This is coding)

Exercise: Converting Binary to Decimal

What is the decimal equivalent of the binary number 1101110?

$$
\begin{aligned}
& 1 \times 2^6 = 1 \times 64 = 64 \\
+ & 1 \times 2^5 = 1 \times 32 = 32 \\
+ & 0 \times 2^4 = 0 \times 16 = 0 \\
+ & 1 \times 2^3 = 1 \times 8 = 8 \\
+ & 1 \times 2^2 = 1 \times 4 = 4 \\
+ & 1 \times 2^1 = 1 \times 2 = 2 \\
+ & 0 \times 2^0 = 0 \times 1 = 0 \\
& = 110 \text{ in base } 10
\end{aligned}
$$

Quick Example

$$25_{10} = 11001_2 = 31_8 = 19_{16}$$

Base

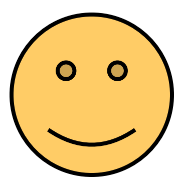
Exercise – Convert ...

Decimal	Binary	Octal	Hexa-decimal
29.8			
	101.1101		
		3.07	
			C.82

Exercise – Convert …

Answer

Decimal	Binary	Octal	Hexa-decimal
29.8	11101.110011…	35.63…	1D.CC…
5.8125	101.1101	5.64	5.D
3.109375	11.000111	3.07	3.1C
12.5078125	1100.10000010	14.404	C.82



Exercise – Convert ...

Decimal	Binary	Octal	Hexa-decimal
33			
	1110101		
		703	
			1AF

Exercise – Convert …

Answer

Decimal	Binary	Octal	Hexa-decimal
33	100001	41	21
117	1110101	165	75
451	111000011	703	1C3
431	110101111	657	1AF