Computer Architecture & Networks

CHAPTER 1 (Part B):
NUMBER SYSTEMS

Learning Outcome

This chapter discusses about number systems.

To understand the conversion of number systems in computational methods.

- Decimal
- Binary
- Hexadecimal
- Octal
- Number conversion
- Binary Coded Decimal

NUMBER SYSTEMS

A set of values used to represent different quantities is known as Number System.

A number system can be used to represent the number of students in a class or number of presenters in a session and etc.

The digital computer represents all kind of data and information in binary numbers.

It includes audio, graphics, video, text and numbers. The total number of digits used in a number system is

System	Base	Symbols
Decimal	10	0, 1, … 9
Binary	2	0, 1
Octal	8	0, 1, 2, …7
Hexa-decimal	16	0, 1, … 9, A, B, … F

Decimal Numbers

The **base** or **radix** of decimal numbers is **10**, because only ten symbols (0 through 9) are used.

The column weights of decimal numbers are powers of ten that increase from right to left

$$10^2 \ 10^1 \ 10^0. \ 10^{-1} \ 10^{-2} \ 10^{-3} \ 10^{-4} \ \dots$$

Example 1:   853 is a decimal number

$$853_{10} = (8 \times 10^2) + (5 \times 10^1) + (3 \times 10^0)$$

Example 2:   9781.024 is a decimal number

$$9781.024_{10} = (9 \times 10^3) + (7 \times 10^2) + (8 \times 10^1) + (1 \times 10^0) +$$

$$(0 \times 10^{-1}) + (2 \times 10^{-2}) + (4 \times 10^{-3})$$

Decimal Numbers

The value of a number is calculated by multiplying each digit by its weight, then add up the totals.

Digit	1	0	2	4	.	5	7
Weight	1000	100	10	1		0.1	0.01
Value	1000 + 0	+ 20 +	4	.	0.5 + 0.07		

Decimal System

Test Yourself:

Express the decimal number 2745.214 as a sum of the values of each digit.

Decimal System

weights 10^3 10^2 10^1 10^0 $10^{-1}$$10^{-2}$$10^{-3}$

2	7	4	5	.	2	1	4

2745.214_{10}

$= (2 \times 10^3) + (7 \times 10^2) + (4 \times 10^1) + (5 \times 10^0) + (2 \times 10^{-1}) + (1 \times 10^{-2}) + (4 \times 10^{-3})$

$= (2 \times 1000) + (7 \times 100) + (4 \times 10) + (5 \times 1) + (2 \times 0.1) + (1 \times 0.01) + (4 \times 0.001)$

$= (2000) + (700) + (40) + (5) + (0.2) + (0.01) + (0.004)$

Binary Numbers

For digital systems, the binary number system is used.

Binary has a radix of **2** and uses the digits **0** and **1** to represent quantities.

The column weights of binary numbers are powers of two that increase from right to left.

$$2^2 \ 2^1 \ 2^0 . \ 2^{-1} \ 2^{-2} \ 2^{-3} \ 2^{-4} \ \ldots$$

Example 1:
$$1011_2 = (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$$

Example 2: 1101.01_2
$$= ((1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}))_{10}$$
$$= ((1 \times 8) + (1 \times 4) + (0 \times 2) + (1 \times 1) + (0 \times 0.5) + (1 \times 0.25))_{10}$$
$$= (8 + 4 + 0 + 1 + 0 + 0.25)_{10}$$
$$= 13.25_{10}$$

POSITIVE POWERS OF TWO (WHOLE NUMBERS)									NEGATIVE POWERS OF TWO (FRACTIONAL NUMBER)					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
256	128	64	32	16	8	4	2	1	1/2	1/4	1/8	1/16	1/32	1/64
									0.5	0.25	0.125	0.0625	0.03125	0.015625

Binary-to-Decimal Conversion

The value of a number is calculated by multiplying each digit by its weight, then add up the totals.

Digit	1	0	1	1	.	1	0	1
Weight	8	4	2	1		0.5	0.25	0.125
Value	8 +	0 +	2 +	1	.	0.5	+ 0 +	0.125

11.625

Convert the binary number 10011.011_2 to decimal.

Example

Convert the binary number 10011.011_2 to decimal.

Solution

Note the subscript 2 clarifies the base of the number (binary).Write out the bits and their weights. Multiply the bit by its corresponding weight and record the result. At the end, add the results to get the decimal number.

Binary	**1**	**0**	**0**	**1**	**1**	**.**	**0**	**1**	**1**
Weights	16	8	4	2	1	**.**	0.5	0.25	0.125

----

16 + 0 + 0 + 2 + 1 + 0 + 0.25 + 0.125

Decimal   **19.375**

Decimal-to-Binary Conversion

▶ Whole decimal numbers are converted into binary as follows:

▶ 135_{10} from decimal into binary. Note the subscript 10 clarifies the base of the number (decimal).
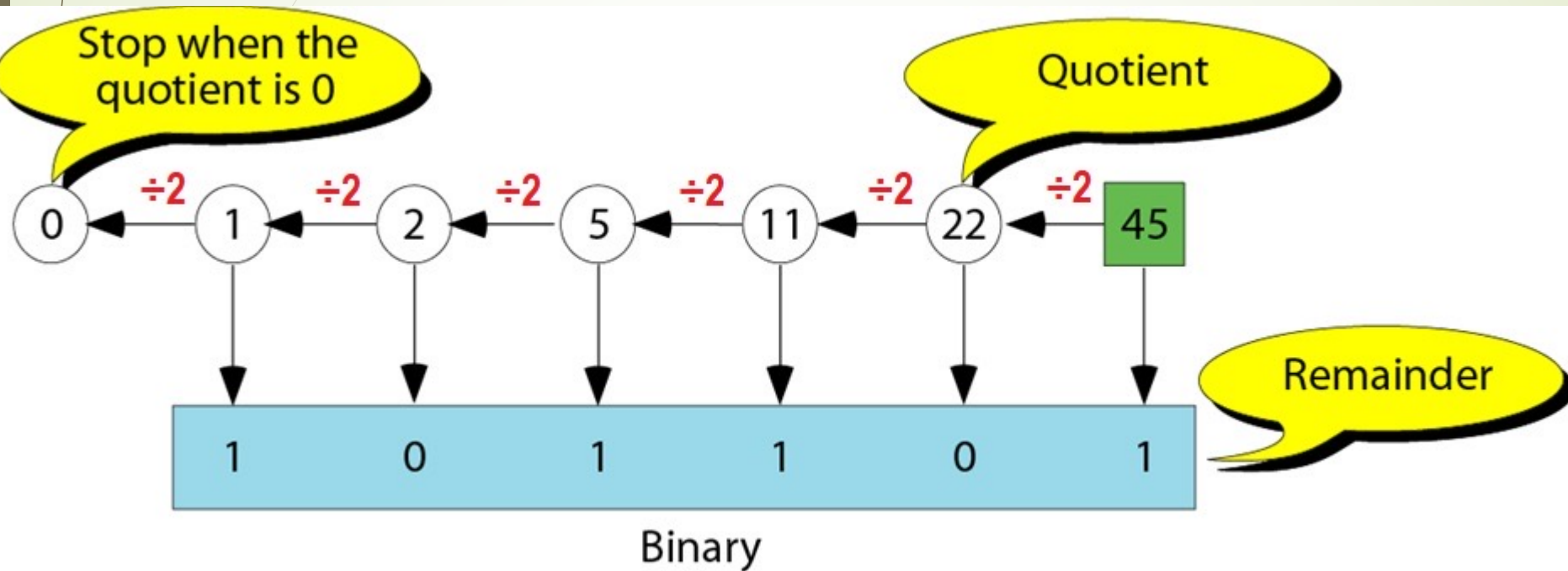
4 + 2 + 1 = *135*

MSB

LSB

128	64	32	16	8	4	2	1
1	0	0	0	0	1	1	1

Decimal-to-Binary Conversion

Integer part: Divide by the base (**2**) and keep track of remainder

Decimal-to-Binary Conversion
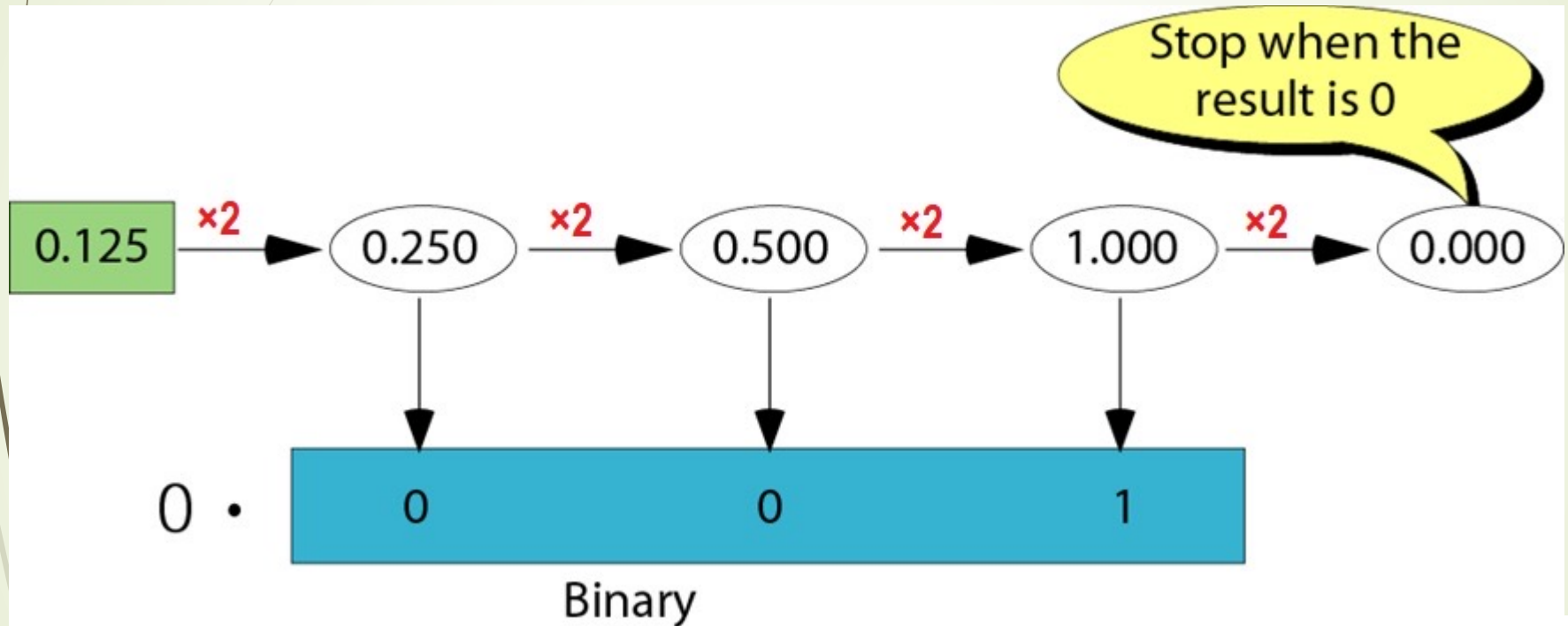
- Fractional part: Multiply by the base (2) and keep track of integer part

Decimal to Binary Conversion

```
2 | 46   (decimal)
2 | 23      remainder   0
2 | 11      remainder   1
2 |  5      remainder   1
2 |  2      remainder   1
2 |  1      remainder   0
     0      remainder   1
```

i.e. $46_{10} = 101110_2$

Convert the decimal number 35.875 to binary.

Example

Convert the decimal number 35 to binary.

Solution

Write out the number at the right corner. Divide the number continuously by 2 and write the quotient and the remainder. The quotients move to the left, and the remainder is recorded under each quotient. Stop when the quotient is zero.

0 ← 1 ← 2 ← 4 ← 8 ← 17 ← 35 Dec.

Binary   1   0   0   0   1   1

Convert the fraction 0.875 to binary

Solution

Write the fraction at the left corner. Multiply the number continuously by 2 and extract the integer part as the binary digit. Stop when the number is 0.0.

0.875 ➔ 1.750 ➔ 1.5 ➔ 1.0 ➔ 0.0

0 . 1 1 1

Thus, the final solution is:

35.875 = 100011.111

Binary Numbers

▶ It is difficult to read long strings of binary numbers-- and even a modestly-sized decimal number becomes a very long binary number.

For example: $11010100011011_2 = 13595_{10}$

▶ For compactness and ease of reading, binary values are usually expressed using the **hexadecimal, or base-16**, numbering system.

Hexadecimal Numbers
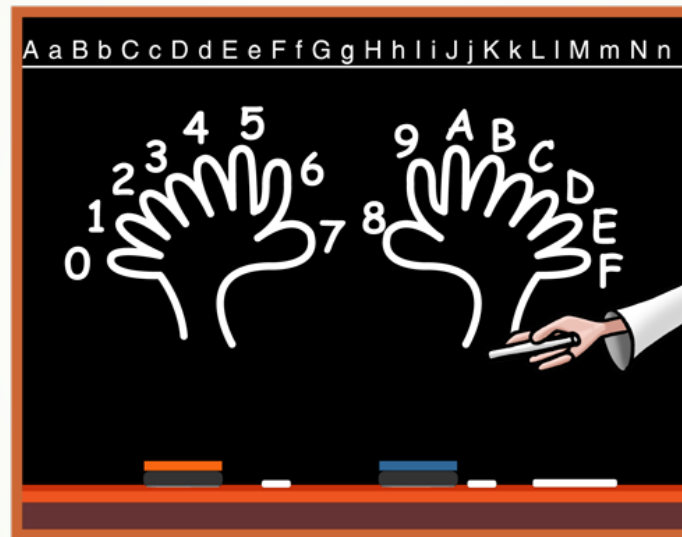
Hexadecimal Numbers

Hexadecimal uses sixteen characters to represent numbers: the numbers 0 through 9 and the alphabetic characters A through F. All modern computing platforms use 32-, or 64-bit words

Large binary number can easily be converted to hexadecimal by grouping 4 bits at a time and writing the equivalent hexadecimal character.

Example Express 1001 0110 0000 1110_2 in hexadecimal:

Solution Group the binary number by 4-bits starting from the right. Thus, 960E

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	0001 0000

Hexadecimal ↔ Binary

A 4-bit binary number can be represented by a hexadecimal digit, and vice versa.

Binary	Hex Digit	Binary	Hex Digit
0000	0	1000	8
0001	1	1001	9
0010	2	1010	A
0011	3	1011	B
0100	4	1100	C
0101	5	1101	D
0110	6	1110	E
0111	7	1111	F

1 1 1 1	1 1 0 0	1 1 1 0	0 1 0 0	Hexadecimal
F	C	E	4	

Convert the following binary number to hexadecimal

11010101000.1111010111

Convert the following binary number to hexadecimal 11010101000.1111010111

Solution

For the integer part, group each 4 bits starting from the first bit to the right of the binary point (LSB) and add any required number of zeros to the left. For the fraction part, group each 4 bits starting from the first bit to the left of the binary point and add any required number of zeros to the right.

$$(\; 0110 \; | \; 1010 \; | \; 1000 \; | . | \; 1111 \; | \; 0101 \; | \; 1100 \;)_2$$

$$(\quad 6 \qquad A \qquad 8 \quad . \quad F \qquad 5 \qquad C \;)_{16}$$

Hexadecimal ↔ Decimal

Hexadecimal is a weighted number system. The column weights are powers of 16, which increase from right to left.

Digit weights $\left\{\begin{array}{cccc} 16^3 & 16^2 & 16^1 & 16^0 \\ 4096 & 256 & 16 & 1 \end{array}\right.$

Example

Express $1A2F_{16}$ in decimal.

Solution

Start by writing the column weights:
4096 256 16 1

  1   A   2 F_{16}

$1(4096) + 10(256) + 2(16) + 15(1) = 6703_{10}$

Decimal	Hexadecimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	0001 0000

Decimal to Hex Conversion

```
16 | 1982₁₀
16 |   123          remainder     14 (E)
 16|      7          remainder     11 (B)
          0          remainder      7
```

i.e. $1982_{10} = 7BE_{16}$

Octal Numbers

Octal uses eight characters the numbers 0 through 7 to represent numbers. It was widely used in computing systems such as IBM mainframes employed 12-bit, 24-bit or 36-bit words. It is still used in file permissions under Unix systems.

Binary number can easily be converted to octal by grouping bits 3 at a time and writing the equivalent octal character for each group.

Example

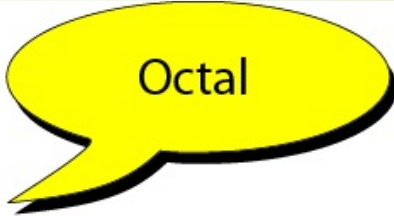Express $1\ 001\ 011\ 000\ 001\ 110_2$ in octal:

Solution

Group the binary number by 3-bits starting from the right. Thus, 113016_8

Decimal	Octal	Binary
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111
8	10	001 000
9	11	001 001
10	12	001 010
11	13	001 011
12	14	001 100
13	15	001 101
14	16	001 110
15	17	001 111
16	20	010 000

Octal ↔ Binary Conversion

*A **3-bit binary number** can be represented by an octal digit, and vice versa.*

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

1	1 1 1	1 1 0	0 1 1	1 0 0	1 0 0	Octal
1	7	6	3	4	4	

Example

Convert the following binary number to octal 11010101000.1111010111

Convert the following binary number to octal

11010101000.1111010111

Solution

For the integer part, group each 3 bits starting from the first bit to the right of the binary point (LSB) and add any required number of zeros to the left. For the fraction part, group each 3 bits starting from the first bit to the left of the binary point and add any required number of zeros to the right.

$$(011|010|101|000.|111|101|011|100)_2$$

$$(\quad 3 \quad 2 \quad 5 \quad 0 \quad . \quad 7 \quad 5 \quad 3 \quad 4 \quad)_8$$

Octal ↔ Decimal

Octal is also a weighted number system. The column weights are powers of 8, which increase from right to left.

Digit weights $\left\{ \begin{array}{cccc} 8^3 & 8^2 & 8^1 & 8^0 \\ 512 & 64 & 8 & 1 \end{array} \right.$

Example Express 3702_8 in decimal.

Solution Start by writing the column weights:
512 64 8 1

3 7 0 2_8

$3(512) + 7(64) + 0(8) + 2(1) =$ 1986_{10}

Decimal	Octal	Binary
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111
8	10	001 000
9	11	001 001
10	12	001 010
11	13	001 011
12	14	001 100
13	15	001 101
14	16	001 110
15	17	001 111
16	20	010 000

Example

Convert the binary number 11010100011011 to hexadecimal and octal numbers.

Convert the binary number 11010100011011 to hexadecimal and octal numbers.

Solution

- Using groups of four bits, the binary number 11010100011011_2 (= 13595_{10}) in hexadecimal is:

0011	0101	0001	1011
3	5	1	B

- Octal (base 8) values are derived from binary by using groups of three bits:

011	010	100	011	011
3	2	4	3	3
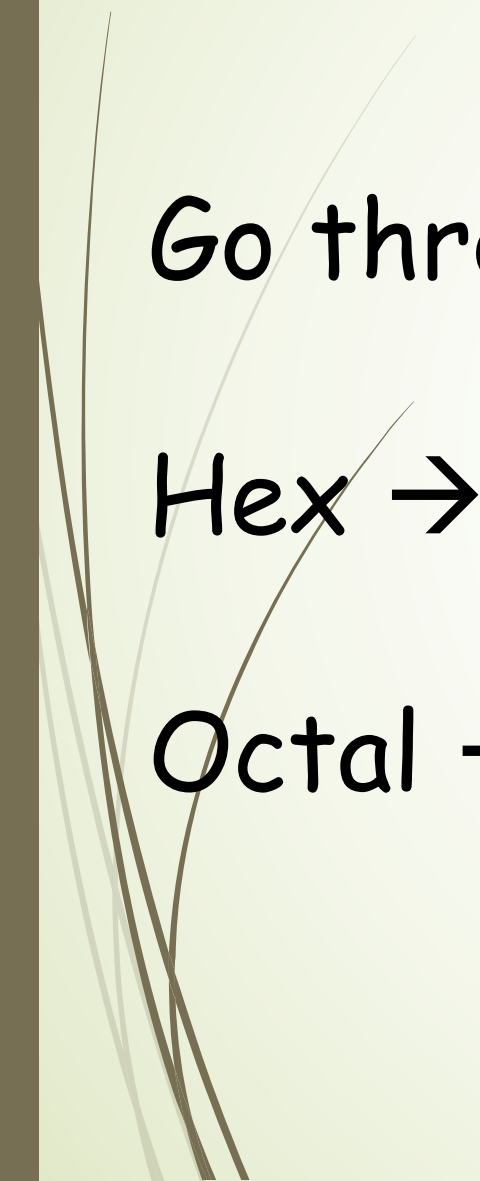
Octal ↔ Hex

Go through Binary/Decimal

Hex → Binary /Decimal → Octal

Octal → Binary /Decimal → Hex

Summary of Number System Conversions
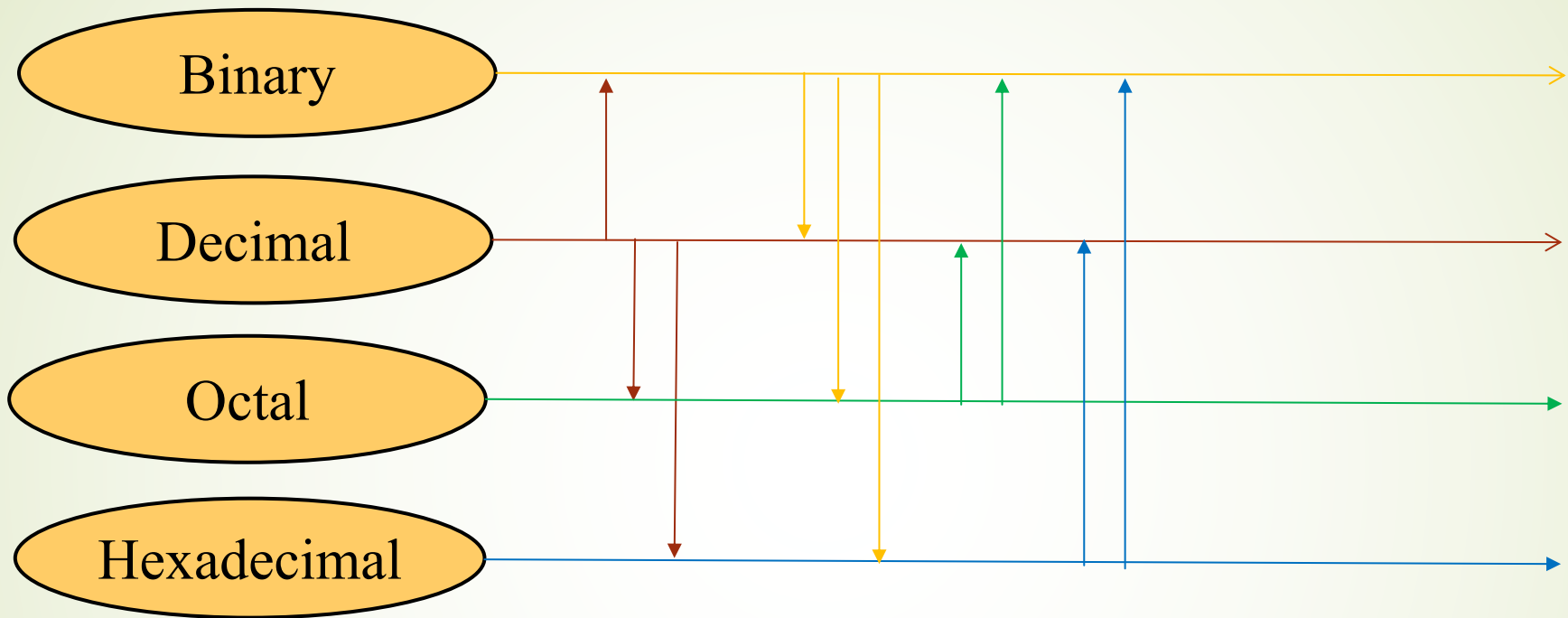
Binary	⇨ Decimal	(sum of all base 2 weighting)
Binary	⇨ Hexadecimal	(4 bits grouping)
Binary	⇨ Octal	(3 bits grouping)
Decimal	⇨ Binary	(divide by 2)
Decimal	⇨ Hexadecimal	(divide by 16)
Decimal	⇨ Octal	(divide by 8)
Hexadecimal	⇨ Binary	(equivalent 4 bits of each hexadecimal digit)
Hexadecimal	⇨ Decimal	(sum of all base 16 weighting)
Octal	⇨ Binary	(equivalent 3 bits of each digit)
Octal	⇨ Decimal	(sum of all base 8 weighting)

Binary-Coded Decimal
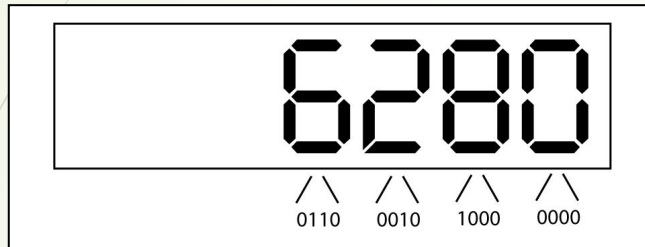
- BCD enables fast conversions from decimal to binary for applications such as pocket calculators.

- Each digit on a calculator corresponds directly to a four-bit block in BCD.

Binary Coded Decimal (BCD)

- BCD represents decimal numbers using blocks of four binary digits.

- Each block of four is converted and the decimal values are then read off:

8	4	2	1
1	0	0	1
8 + 0 + 0 + 1			
9			

8	4	2	1
0	0	1	1
0 + 0 + 2 + 1			
3			

8	4	2	1
1	0	0	0
8 + 0 + 0 + 0			
8			

- Therefore **1001 0011 1000** in BCD = **938** in denary.

Binary-Coded Decimal

➤ To code a number with *n* decimal digits, we need *4n* bits in BCD

e.g. $(365)_{10}$ = $(0011\ 0110\ 0101)_{BCD}$

➤ This is different to converting to binary, which is $(365)_{10}$ = $(101101101)_2$

➤ Clearly, BCD requires more bits. BUT it is easier to understand/interpret

Binary-Coded Decimal

The table illustrates the difference between straight binary and BCD. BCD represents each decimal digit with a 4-bit code.

Notice that the codes 1010 through 1111 are not used in BCD.

Decimal	Binary	BCD
0	0000	0000
1	0001	0001
2	0010	0010
3	0011	0011
4	0100	0100
5	0101	0101
6	0110	0110
7	0111	0111
8	1000	1000
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010
13	1101	0001 0011
14	1110	0001 0100
15	1111	0001 0101

BINARY ADDITION

The rules for binary addition are

$0 + 0 = 0$	Sum = 0, carry = 0
$0 + 1 = 0$	Sum = 1, carry = 0
$1 + 0 = 0$	Sum = 1, carry = 0
$1 + 1 = 10$	Sum = 0, carry = 1

When an input carry = 1 due to a previous result, the rules are

$1 + 0 + 0 = 01$	Sum = 1, carry = 0
$1 + 0 + 1 = 10$	Sum = 0, carry = 1
$1 + 1 + 0 = 10$	Sum = 0, carry = 1
$1 + 1 + 1 = 10$	Sum = 1, carry = 1

BINARY SUBTRACTION

The rules for binary subtraction are

$$0 - 0 = 0$$

$$1 - 1 = 0$$

$$1 - 0 = 1$$

$$10 - 1 = 1 \text{ with a borrow of } 1$$

Binary Addition

$$101101 \qquad (45_{10})$$
$$\underline{111110} + \quad (62_{10})$$
$$1101011 \qquad (107_{10})$$

Binary Subtraction

$$110011 \qquad (51)$$
$$\underline{011101} - \quad (29)$$
$$010110 \qquad (22)$$

Binary Multiplication

$$1101 \quad (13_{10})$$
$$\underline{1011} \text{ x} \quad (11_{10})$$
$$1101$$
$$11010$$
$$000000$$
$$\underline{1101000} \ +$$
$$10001111 \quad (143_{10})$$

BINARY ADDITION – SELF TEST 1

1. 11011 + 1001010 =
2. 1011001 + 111010 =

- 11011 + 1001010 = 1100101:

$$
\begin{array}{ccccccc}
 & 1 & 1 & & & 1 & & \\
 & & & 1 & 1 & 0 & 1 & 1 \\
+ & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
\end{array}
$$

- 1011001 + 111010 = 10010011:

$$
\begin{array}{cccccccc}
 & 1 & 1 & 1 & 1 & & & \\
 & & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\
+ & & & 1 & 1 & 1 & 0 & 1 & 0 \\
\hline
 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\
\end{array}
$$

BINARY SUBTRACTION – SELF TEST 2

1. 1000101 - 101100 =
2. 1110110 - 1010111 =

- $1000101 - 101100 = 11001$:

$$
\begin{array}{ccccccc}
 & 0 & 1 & 1 & & & \\
\not1 & \not1 0 & \not1 0 & {}^1 0 & 1 & 0 & 1 \\
- & 1 & 0 & 1 & 1 & 0 & 0 \\
\hline
 & & 1 & 1 & 0 & 0 & 1 \\
\end{array}
$$

- $1110110 - 1010111 = 11111$:

$$
\begin{array}{ccccccc}
 & 0 & {}^1 0 & 1 & {}^1 0 & {}^1 0 & \\
1 & \not1 & \not1 & \not1 0 & \not1 & \not1 & {}^1 0 \\
- & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\
\hline
 & & 1 & 1 & 1 & 1 & 1 \\
\end{array}
$$

END