# ENS1161 Computer Fundamentals

# Lecture 07

# Bases and Number Representation

Dr Włodzimierz Górnisiewicz – School of Engineering

# Outline

01. The decimal number system
02. Use of subscripts
03. The octal number system
04. Counting in octal
05. Conversion from decimal to octal
06. Conversion of octal to decimal
07. The binary number system
08. Counting in binary
09. Conversion from decimal to binary
10. Conversion from binary to decimal
11. The hexadecimal number system
12. Counting in hexadecimal
13. Conversion of hexadecimal numbers to binary
14. Conversion of binary numbers to hexadecimal
15. Conversion between hexadecimal and decimal
16. Representation of binary fractions
17. Converting decimal fractions to binary
18. Converting binary fractions to decimal
19. Summary of Number Conversion
20. Arithmetic using other number systems
21. Addition in octal
22. Addition in binary
23. Addition in hexadecimal
24. Binary Coded Decimal code (BCD)
25. Addition of BCD numbers

# Lecture's Major Objectives

After completing this section, students should be able to:

- count in each of the decimal, octal, binary and hexadecimal number systems
- convert integers between the number systems
- convert fractions between binary and decimal number systems
- add integers in each of the number systems
- find the BCD code for a decimal number
- perform addition in each of the number systems
- convert a BCD number to its decimal equivalent
- perform BCD additions, making any necessary decimal adjustments

# The decimal number system

We are familiar with the decimal number system, which is based on the number 10 (probably because we have 10 fingers).

The decimal system uses positional notation.

For example

"354" represents $3\times100 + 5\times10 + 4\times1 = 3\times10^2 + 5\times10 + 4\times1$

whereas

"435" represents $4\times100 + 3\times10 + 5\times1$

In the decimal system we use 10 different symbols to represent numbers.

These symbols are: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9

# Use of subscripts

In the following notes, different subscripts may be used to indicate the number system and hence avoid ambiguity.

For example

| | |
|---|---|
| $354_{10}$ | represents a decimal number |
| $354_8$ | represents an octal number |
| $1001010_2$ | represents a binary number |
| $354_{16}$ | represents a hexadecimal number |

# The octal number system

In the octal system, which is also positional,

"354" means $3 \times 8^2 + 5 \times 8 + 4 \times 1$ which is equal to $236_{10}$ and
"435" means $4 \times 8^2 + 3 \times 8 + 5 \times 1$ which is equal to $285_{10}$

Octal uses only the 8 symbols 0, 1, 2, 3, 4, 5, 6 and 7.
Here are some octal numbers and their decimal equivalents:

| Octal number | Meaning | Dec. equivalent |
|---|---|---|
| 35 | 3x8 + 5x1 | 29 |
| 54 | 5x8 + 4x1 | 44 |
| 77 | 7x8 + 7x1 | 63 |
| 261 | $2x8^2$ + 6x8 +1x1 | 177 |

# Counting in octal

| 64 | 8 | 1 | Dec. Equiv |
|---|---|---|---|
| | | 0 | 0 |
| | | 1 | 1 |
| | | 2 | 2 |
| | | ... | ... |
| | | 7 | 7 |
| | 1 | 0 | 8 |
| | 1 | 1 | 9 |
| | ... | ... | ... |
| | 1 | 7 | 15 |
| | 2 | 0 | 16 |
| | 2 | 1 | 17 |
| | ... | ... | ... |
| | 7 | 7 | 63 |
| 1 | 0 | 0 | 64 |
| 1 | 0 | 1 | 65 |
| ... | ... | ... | ... |
| 1 | 7 | 7 | 127 |
| 2 | 0 | 0 | 128 |

# Conversion from decimal to octal

The algorithm is:

"Divide by 8 until the answer is 0, then read the remainders in reverse order“

Example: Convert $171_{10}$ to octal

When we divide 171 repeatedly by 8, we get:

| | |
|---|---|
| **8** | **171** |
| **8** | **21 + rem 3** |
| **8** | **2 + rem 5** |
| **8** | **0 + rem 2** |

The remainders, in reverse order, are: 2, 5 and 3.

So $171_{10} = 253_8$

Why does this method work?

$171 = 21 \times 8 + 3$ and

$21 = 2 \times 8 + 5$, so

$171 = [(2 \times 8) + 5] \times 8 + 3$

therefore $171 = 2 \times 8^2 + 5 \times 8 + 3$

# Conversion from decimal to octal

Example: Convert $278_{10}$ to octal

| | |
|---|---|
| 8 | 278 |
| 8 | 34 + rem 6 |
| 8 | 4 + rem 2 |
| 8 | 0 + rem 4 |

So $278_{10} = 426_8$

[ Check: $426_8 = 4 \times 64 + 2 \times 8 + 6 \times 1 = 278_{10}$ ]

# Conversion of octal to decimal

To convert from octal to decimal, we simply recall that each octal digit represents a particular power of 8.

Example 1: Convert $213_8$ to decimal

$$213_8 = 2 \times 8^2 + 1 \times 8 + 3 \times 1$$
$$= 2 \times 64 + 1 \times 8 + 3 \times 1 = 139_{10}$$

Example 2: Convert $467_8$ to decimal

$$467_8 = 4 \times 8^2 + 6 \times 8 + 7 \times 1$$
$$= 4 \times 64 + 6 \times 8 + 7 \times 1 = 311_{10}$$

# Conversion of octal to decimal

Example 3: Convert $4276_8$ to decimal

$$\begin{aligned} 4276_8 &= 4 \times 8^3 + 2 \times 8^2 + 7 \times 8 + 6 \times 1 \\ &= 4 \times 512 + 2 \times 64 + 7 \times 8 + 6 \times 1 \\ &= 2238_{10} \end{aligned}$$

In fact there is a more convenient way to evaluate longer expressions such as in Example 3, especially if you are using a calculator:

$$4276_8 = (((4 \times 8) + 2) \times 8 + 7) \times 8 + 6 = 2238_{10}$$

# The binary number system

The binary number system is based on the number 2. It also uses positional notation. In this system there are only two symbols, 0 and 1.

So, for example, the binary number 101 represents
$1 \times 2^2 + 0 \times 2 + 1 \times 1 = 5$ in decimal
and the binary number 1101 represents
$1 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1 \times 1 = 13$ in decimal

Here are some binary numbers and their decimal equivalents:

| Binary number | Meaning | Dec. equivalent |
|---|---|---|
| 111 | 4+2+1 | 7 |
| 1001 | 8+0+0+1 | 9 |
| 110011 | 32+16+0+0+2+1 | 51 |
| 10000000 | 128+0+...+0 | 128 |

# Counting in binary

| 16 | 8 | 4 | 2 | 1 | Dec. equiv |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 | 2 |
| 0 | 0 | 0 | 1 | 1 | 3 |
| 0 | 0 | 1 | 0 | 0 | 4 |
| 0 | 0 | 1 | 0 | 1 | 5 |
| 0 | 0 | 1 | 1 | 0 | 6 |
| 0 | 0 | 1 | 1 | 1 | 7 |
| 0 | 1 | 0 | 0 | 0 | 8 |
| 0 | 1 | 0 | 0 | 1 | 9 |
| 0 | 1 | 0 | 1 | 0 | 10 |
| 0 | 1 | 0 | 1 | 1 | 11 |
| 0 | 1 | 1 | 0 | 0 | 12 |
| 0 | 1 | 1 | 0 | 1 | 13 |
| 0 | 1 | 1 | 1 | 0 | 14 |
| 0 | 1 | 1 | 1 | 1 | 15 |
| 1 | 0 | 0 | 0 | 0 | 16 |
| 1 | 0 | 0 | 0 | 1 | 17 |
| 1 | 0 | 0 | 1 | 0 | 18 |
| 1 | 0 | 0 | 1 | 1 | 19 |
| 1 | 0 | 1 | 0 | 0 | 20 |
| 1 | 0 | 1 | 0 | 1 | 21 |
| 1 | 0 | 1 | 1 | 0 | 22 |
| 1 | 0 | 1 | 1 | 1 | 23 |
| 1 | 1 | 0 | 0 | 0 | 24 |
| 1 | 1 | 0 | 0 | 1 | 25 |
| 1 | 1 | 0 | 1 | 0 | 26 |
| 1 | 1 | 0 | 1 | 1 | 27 |
| 1 | 1 | 1 | 0 | 0 | 28 |
| 1 | 1 | 1 | 0 | 1 | 29 |
| 1 | 1 | 1 | 1 | 0 | 30 |
| 1 | 1 | 1 | 1 | 1 | 31 |

# Conversion from decimal to binary

Method 1:

Divide by 2 until the answer is 0, then read the remainders in reverse order.

Example 1: Convert $58_{10}$ to binary.

| | |
|---|---|
| **2** | **58** |
| **2** | **29 + rem 0** |
| **2** | **14 + rem 1** |
| **2** | **7 + rem 0** |
| **2** | **3 + rem 1** |
| **2** | **1 + rem 1** |
| | **0 + rem 1** |

The remainders, in reverse order, are 1, 1, 1, 0, 1 and 0.

Therefore $58_{10} = 111010_2$

[ Check: $111010_2 = 32 + 16 + 8 + 0 + 2 + 0 = 58_{10}$ ]

# Conversion from decimal to binary

An alternative (and much quicker!) method is to use octal.

First you need to know the binary equivalents of the numbers 0, 1, 2, …, 7.

These are:

| | |
|---|---|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

# Conversion from decimal to binary

Method 2:

First convert the decimal number to octal, and then replace each octal digit by its binary equivalent.

Example 1 (again): Convert $58_{10}$ to binary

First convert to octal:

| | |
|---|---|
| **8** | **58** |
| **8** | **7 + rem 2** |
| **8** | **0 + rem 7** |

Reading the remainders in reverse order, we find $58_{10} = 72_8$

Now replace each octal digit with the corresponding binary equivalent.

(In this case replace 7 by 111 and replace 2 by 010)

So $58_{10} = 72_8 = 111010_2$

# Conversion from decimal to binary

Example 2: Convert $370_{10}$ to binary.

First convert to octal:

| | |
|---|---|
| **8** | **370** |
| **8** | **46 + rem 2** |
| **8** | **5 + rem 6** |
| **8** | **0 + rem 5** |

Now replace each octal digit with the corresponding binary equivalent.

So $370_{10} = 562_8 = 101110010_2$

# Conversion from binary to decimal

Method 1:

Add up the powers of 2

Example 1: Convert $10111001_2$ to decimal

$$10111001_2 = 2^7 + 2^5 + 2^4 + 2^3 + 1$$
$$= 128 + 32 + 16 + 8 + 1 = 185_{10}$$

Example 2: Convert $101111011_2$ to decimal

$$101111011_2 = 2^8 + 2^6 + 2^5 + 2^4 + 2^3 + 2 + 1$$
$$= 256 + 64 + 32 + 16 + 8 + 2 + 1 = 379_{10}$$

Example 3: Convert $111100_2$ to decimal

$$111100_2 = 2^5 + 2^4 + 2^3 + 2^2$$
$$= 32 + 16 + 8 + 4 = 60_{10}$$

# Conversion from binary to decimal

Method 2

Convert to octal first, then to decimal.
Separate groups of three binary digits, starting at the units digit. Then replace each group of three binary digits by its octal equivalent.

| | |
|---|---|
| 0 | 000 |
| 1 | 001 |
| 2 | 010 |
| 3 | 011 |
| 4 | 100 |
| 5 | 101 |
| 6 | 110 |
| 7 | 111 |

# Conversion from binary to decimal

Example 1 (again): Convert $10111001_2$ to decimal

$10 \mid 111 \mid 001 = 271_8 = 2 \times 64 + 7 \times 8 + 1 = 185_{10}$

2 7 1

Example 2 (again): Convert $101111011_2$ to decimal

$101\ 111\ 011 = 573_8 = 5 \times 64 + 7 \times 8 + 3 = 379_{10}$

Example 3 (again): Convert $111100_2$ to decimal

$111\ 100 = 74_8 = 7 \times 8 + 4 = 60_{10}$

# The hexadecimal number system

This system is based on the number 16. Its main purpose is to help us to cope with long strings of binary digits. For example suppose we had to remember the binary number:

00101101111100111000110001001011

We would find it much easier to work with its hexadecimal equivalent

2DF38C4B

| $16^3$ | $16^2$ | 16 | 1 | Dec. equiv. |
|---|---|---|---|---|
| | | 1 | 5 | 21 |
| | 1 | 0 | 3 | 259 |
| | | 5 | 7 | 87 |
| 1 | 0 | 0 | 0 | 4096 |
| | | 4 | 6 | 70 |

# The hexadecimal number system

Hexadecimal uses 16 symbols. The symbols 0, 1, 2, …, 9 have the same meaning as in decimal, and we need six extra symbols to represent these six numbers.
We use A, B, C, D, E and F:

A represents the decimal number 10,
B represents the decimal number 11,
C represents the decimal number 12,
D represents the decimal number 13,
E represents the decimal number 14, and
F represents the decimal number 15.

"5C" in hex represents $5 \times 16 + 12 = 92$ in decimal
"1AF" in hex represents $1 \times 16^2 + 10 \times 16 + 15 \times 1 = 431$ in decimal
$345_{16} = 837_{10}$ and $1AF_{16} = 431_{10}$

# The hexadecimal number system

Here are some more hexadecimal numbers and their decimal equivalents:

(Note: $16^2 = 256$ and $16^3 = 4096$)

| Hex. number | Meaning | Dec. equiv. |
|---|---|---|
| 2B7 | 2x256 + 11x16 + 7x1 | 695 |
| 496 | 4x256 + 9x16 + 6x1 | 1174 |
| FE | 15x16 + 14x1 | 254 |
| CB4D | 12x4096 + 11x256 + 4x16 +13x1 | 52045 |

# Counting in hexadecimal

| $16^3$ | $16^2$ | 16 | 1 | Dec. equiv. |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 2 | 2 |
| ... | ... | ... | ... | ... |
| 0 | 0 | 0 | 9 | 9 |
| 0 | 0 | 0 | A | 10 |
| ... | ... | ... | ... | ... |
| 0 | 0 | 0 | F | 15 |
| 0 | 0 | 1 | 0 | 16 |
| 0 | 0 | 1 | 1 | 17 |
| ... | ... | ... | ... | ... |
| 0 | 0 | 1 | D | 29 |
| 0 | 0 | 1 | E | 30 |
| 0 | 0 | 1 | F | 31 |
| 0 | 0 | 2 | 0 | 32 |
| ... | ... | ... | ... | ... |
| 0 | 0 | 9 | F | 159 |
| 0 | 0 | A | 0 | 160 |
| ... | ... | ... | ... | ... |
| 0 | 0 | F | F | 255 |
| 0 | 1 | 0 | 0 | 256 |
| ... | ... | ... | ... | ... |
| F | F | F | F | 65535 |

# Conversion of hexadecimal numbers to binary

First we need to be familiar with the binary equivalents of the hexadecimal numbers 0, …, F. These are:

| | |
|---|---|
| **0** | **0000** |
| **1** | **0001** |
| **2** | **0010** |
| **3** | **0011** |
| **4** | **0100** |
| **5** | **0101** |
| **6** | **0110** |
| **7** | **0111** |

| | |
|---|---|
| **8** | **1000** |
| **9** | **1001** |
| **A** | **1010** |
| **B** | **1011** |
| **C** | **1100** |
| **D** | **1101** |
| **E** | **1110** |
| **F** | **1111** |

Then it is simply a matter of replacing each hexadecimal digit by the corresponding four binary digits.

# Conversion of binary numbers to hexadecimal

To convert a binary integer to hexadecimal, separate binary digits into groups of four, starting at the units digit, and then replace each group by its hexadecimal equivalent.

Example 1:

Convert $101111101010110011_2$ to hexadecimal.

$$10\ \ 1111\ \ 1010\ \ 1011\ \ 0011_2 = 2FAB3_{16}$$

Example 2:

Convert $1111100001011101101_2$ to hexadecimal.

$$111\ \ 1100\ \ 0010\ \ 1110\ \ 1101_2 = 7C2ED_{16}$$

# Conversion between hexadecimal and decimal

Sometimes it is convenient to be able to convert from simple hexadecimal numbers to decimal, or vice versa.

For conversions such as these it helps to know, or at least be familiar with, the multiples of 16.

Examples:

1. The hex number 3C means $3 \times 16 + 12 = 60$ in decimal.
2. The hex number 54 means $5 \times 16 + 4 = 84$ in decimal.
3. The hex number F6 means $15 \times 16 + 6 = 246$ in decimal.

# Conversion between hexadecimal and decimal

4. To convert the decimal number 43 to hexadecimal:

   43 divided by 16?
   $2 \times 16 = 32$, so
   43 divided by 16 gives 2 with 11 remainder
   So $43 = 2 \times 16 + 11$

   Therefore 43 in decimal becomes 2B in hex.

5. To convert the decimal number 77 to hexadecimal:

   77 divided by 16?
   $4 \times 16 = 64$, so
   77 divided by 16 gives 4 with 13 remainder
   So $77 = 4 \times 16 + 13$

   Therefore 77 in decimal becomes 4D in hex.

# Conversion between hexadecimal and decimal

6. To convert the decimal number 103 to hexadecimal:

   103 divided by 16?
   $6 \times 16 = 96$, so
   103 divided by 16 gives 6 with 7 remainder
   So $103 = 6 \times 16 + 7$

   Therefore 103 in decimal becomes 67 in hex.

# Representation of binary fractions

Now we consider fractions in binary. The concepts are analogous to those in the decimal system.

In the decimal system, the columns to the right of the decimal point represent

$$\frac{1}{10}, \quad \frac{1}{100}, \quad \frac{1}{1000}, \ldots \text{ and so on.}$$

Here are some decimal numbers with both integer parts and fractional parts:

| ... | 100 | 10 | 1 | | $\frac{1}{10}$ | $\frac{1}{100}$ | $\frac{1}{1000}$ | ... | Meaning |
|---|---|---|---|---|---|---|---|---|---|
| | | 4 | 3 | . | 2 | 5 | | | 43.25 |
| | | | 6 | . | 0 | 7 | 1 | | 6.071 |
| | 9 | 8 | 5 | . | 3 | 0 | 2 | | 985.302 |
| | | | 0 | . | 9 | 9 | 0 | | 0.99 |
| | | | 0 | . | 3 | 2 | 4 | | 0.324 |

# Representation of binary fractions

In decimal, for example

$$0.324 \text{ means } \mathbf{3x\,\frac{1}{10} + 2x\,\frac{1}{100} + 4x\,\frac{1}{1000}} \text{ or } \mathbf{\frac{324}{1000}}$$

Notice that the fraction 0.324 has 3 places, and so the last place

represents $\mathbf{\frac{1}{10^3}}$

In the binary system, the columns to the right of the point represent

$\mathbf{\frac{1}{2}}$ , $\mathbf{\frac{1}{4}}$ , $\mathbf{\frac{1}{8}}$ , $\mathbf{\frac{1}{16}}$ , … and so on.

# Representation of binary fractions

Here are some binary numbers with both integer parts and fractional parts:

| ... | 4 | 2 | 1 | | $\frac{1}{2}$ | $\frac{1}{4}$ | $\frac{1}{8}$ | $\frac{1}{16}$ | ... | Binary meaning | Decimal meaning |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | . | 1 | | | | | 1.1 | 1.5 |
| | | | 0 | . | 1 | 1 | | | | 0.11 | 0.75 |
| | | 1 | 1 | . | 0 | 1 | | | | 11.01 | 3.25 |
| | 1 | 0 | 1 | . | 1 | 1 | 1 | | | 101.111 | 5.875 |
| | | | 0 | . | 1 | 1 | 0 | 1 | | 0.1101 | 0.8125 |

# Representation of binary fractions

In the binary system, for example

$0.1101_2$ means $\frac{1}{2} + \frac{1}{4} + \frac{0}{8} + \frac{1}{16}$ or $\frac{13}{16}$ = 0.8125 in decimal

Here the fraction 0.1101 has 4 places, and so the last place represents or

As another example

$0.10011_2$ means $\frac{1}{2} + \frac{1}{16} + \frac{1}{32}$ or $\frac{19}{32}$ or 0.59375

# Converting decimal fractions to binary

The usual method is:

Repeatedly multiply the digits to the right of the decimal point by 2, until only zeros remain. Then read the integer parts of the successive answers.

Example 1: Convert the decimal fraction 0.34375 to binary

$$
\begin{array}{r}
0.34375 \\
\underline{\quad\quad\quad 2} \\
0.68750 \\
\underline{\quad\quad\quad 2} \\
1.37500 \\
\underline{\quad\quad\quad 2} \\
0.75000 \\
\underline{\quad\quad\quad 2} \\
1.50000 \\
\underline{\quad\quad\quad 2} \\
1.00000
\end{array}
$$

So $0.34375_{10} = 0.01011_2$

# Converting decimal fractions to binary

Example 2: Convert the decimal fraction 0.453125 to binary

**0.453125**
**× 2**

**0.906250**
**× 2**

**1.812500**
**× 2**

**1.625000**
**× 2**

**1.250000**
**× 2**

**0.500000**
**× 2**

**1.000000**

So $0.453125_{10} = 0.011101_2$

# Converting decimal fractions to binary

Example 3: Convert the decimal fraction 0.65625 to binary

$$
\begin{array}{r}
0.65625 \\
\underline{\qquad\qquad 2} \\
\mathbf{1}.31250 \\
\underline{\qquad\qquad 2} \\
\mathbf{0}.62500 \\
\underline{\qquad\qquad 2} \\
\mathbf{1}.25000 \\
\underline{\qquad\qquad 2} \\
\mathbf{0}.50000 \\
\underline{\qquad\qquad 2} \\
\mathbf{1}.00000
\end{array}
$$

So $0.65625_{10} = 0.10101_2$

# Converting binary fractions to decimal

A simple method is:

Ignore the point and convert the fractional part to decimal number. Then divide this by the appropriate power of 2 as given by the number of places in the fractional part.

Example 1: Convert $0.110011_2$ to decimal.

Ignoring the point, 110011 in binary is 51 in decimal.
Now the fraction has 6 places, so we must divide by $2^6 = 64$

Therefore $0.110011_2 = \dfrac{\mathbf{51}}{\mathbf{64}} = 0.796875_{10}$

# Converting binary fractions to decimal

Example 2: Convert $0.11101_2$ to decimal.

Ignoring the point, 11101 in binary is 29 in decimal.
Now the fraction has 5 places, so we must divide by $2^5 = 32$

Therefore $0.11101_2 = \dfrac{29}{32} = 0.90625_{10}$

Example 3: Convert $0.001010101_2$ to decimal.

1010101 in binary is 85 in decimal
The fraction has 9 places, so we must divide by $2^9 = 512$

Therefore $0.001010101_2 = \dfrac{85}{512} = 0.166015625_{10}$

# Converting binary fractions to decimal

Example 4: Convert $0.00000101_2$ to decimal.

101 in binary is 5 in decimal

The fraction has 8 places, so we must divide by $2^8 = 256$

Therefore $0.00000101_2 = \dfrac{\mathbf{5}}{\mathbf{256}} = 0.01953125_{10}$

# Summary of number conversion

| *Conversion* | *Method* | *Example* |
|---|---|---|
| **Binary to** | | |
| Octal | Substitution | $10111011001_2 = 10\ 111\ 011\ 001_2 = 2731_8$ |
| Hexadecimal | Substitution | $10111011001_2 = 101\ 1101\ 1001_2 = 5D9_{16}$ |
| Decimal | Summation | $10111011001_2 = 1x1024 + 0x512 + 1x256 + 1x128 + 1x64 + 0x32 + 1x16 + 1x8 + 0x4 + 0x2 + 1x1 = 1497_{10}$ |
| **Octal to** | | |
| Binary | Substitution | $1234_8 = 001\ 010\ 011\ 100_2$ |
| Hexadecimal | Substitution | $1234_8 = 001\ 010\ 011\ 100_2 = 0010\ 1001\ 1100_2 = 29C_{16}$ |
| Decimal | Summation | $1234_8 = 1x512 + 2x64 + 3x8 + 4x1 = 668_{10}$ |

# Summary of number conversion

| *Conversion* | *Method* | *Example* |
|---|---|---|
| **Hexadecimal to** | | |
| Binary | Substitution | $C0DE_{16} = 1100\ 0000\ 1101\ 1110_2$ |
| Octal | Substitution | $C0DE_{16} = 1100\ 0000\ 1101\ 1110_2$<br>$= 1\ 100\ 000\ 011\ 011\ 110_2 = 140336_8$ |
| Decimal | Summation | $C0DE_{16} = 12x4096 + 0x256 + 13x16 + 14x1 = 49374_{10}$ |

# Summary of number conversion

| *Conversion* | *Method* | *Example* |
|---|---|---|
| **Decimal to** | | |
| Binary | Division | $108_{10} \div 2 = 54$ reminder 0 (LSB)<br>$\div 2 = 27$ reminder 0<br>$\div 2 = 13$ reminder 1<br>$\div 2 = 6$ reminder 1<br>$\div 2 = 3$ reminder 0<br>$\div 2 = 1$ reminder 1<br>$\div 2 = 0$ reminder 1 (MSB)<br><br>$108_{10} = 1101100_2$ |
| Octal | Division | $108_{10} \div 8 = 13$ reminder 4 (LSB)<br>$\div 8 = 1$ reminder 5<br>$\div 8 = 0$ reminder 1 (MSB)<br><br>$108_{10} = 154_8$ |
| Hexadecimal | Division | $108_{10} \div 16 = 6$ reminder 12 (LSB)<br>$\div 16 = 0$ reminder 6 (MSB)<br><br>$108_{10} = 6C_{16}$ |

# Arithmetic using other number systems

We are mainly concerned with the addition of two numbers. An example in the decimal system:

$$
\begin{array}{cccc}
\mathbf{1} & \mathbf{1} & & \\
2 & 8 & 5 & \\
4 & 3 & 8 & + \\
\hline
7 & 2 & 3 &
\end{array}
$$

In 1st column, 8 + 5 = 13, so we "put down 3 and carry is 1"
In 2nd column, 3 + 8 + 1 = 12, we "put down 2 and carry is 1"

# Addition in octal

Remember that in octal the digits represent 1, 8, $8^2$, $8^3$, and so on.
So, if the sum of the two digits in the 1's column is 8 or more, there will be a carry of 1 into the next column, and so on.

Example 1 :

$$
\begin{array}{cccc}
\mathbf{1} & \mathbf{1} & & \\
2 & 7 & 5 & \\
3 & 6 & 7 & + \\
\hline
6 & 6 & 4 &
\end{array}
$$

In 1st column, $7_8 + 5_8 = 14_8$ , so the carry is 1
In 2nd column, $6_8 + 7_8 + 1_8 = 16_8$, so the carry is 1

# Addition in octal

Example 2:

$$\begin{array}{ccccc} 1 & 1 & 1 & & \\ 4 & 5 & 3 & 6 & \\ 2 & 2 & 4 & 5 & + \\ \hline 7 & 0 & 0 & 3 & \end{array}$$

In 1st column, $6_8 + 5_8 = 13_8$ , so the carry is 1
In 2nd column, $4_8 + 3_8 + 1_8 = 10_8$ , so the carry is 1
In 3rd column, $2_8 + 5_8 + 1_8 = 8_8$, so the carry is 1

# Addition in binary

Example 1:

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | | |
| | 1 | 0 | 1 | 1 | |
| | | 1 | 0 | 1 | + |
| 1 | 0 | 0 | 0 | 0 | |

In 1st column, 1 + 1 = 10, so the carry is 1
In 2nd column, 0 +1 + 1 = 10, so the carry is 1
In 3rd column, 1 + 0 + 1 = 10 , so the carry is 1
In 4th column, 1 + 1 = 10 , so the carry is 1

# Addition in binary

Example 2:

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | | |
| | 1 | 1 | 1 | 0 | 1 | |
| | 1 | 0 | 1 | 1 | 1 | + |
| 1 | 1 | 0 | 1 | 0 | 0 | |

In 1st column, 1 + 1 = 10, so the carry is 1
In 2nd column, 1 + 0 + 1 = 10, so the carry is 1
In 3rd column, 1 + 1 + 1 = 11 , so the carry is 1
In 4th column, 0 + 1 + 1 = 10, so the carry is 1
In 5th column, 1 + 1 + 1 = 11 , so the carry is 1

# Addition in binary

Example 3:

$$
\begin{array}{cccccccccc}
 & \color{blue}{1} & \color{blue}{1} & \color{blue}{1} & \color{blue}{1} & \color{blue}{1} & & & & \\
1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & \\
 & & 1 & 0 & 1 & 1 & 1 & 1 & 0 & + \\
\hline
1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 &
\end{array}
$$

# Addition in hexadecimal

In hexadecimal the digits represent 1, 16, $16^2$, 163, and so on.
So, if the sum of the two digits in the 1's column is 16 or more, there will be a carry of 1 into the next column, and so on.

Examples:

$$\begin{array}{cccc} & 6 & 1 & \\ & 3 & 7 & + \\ \hline & 9 & 8 & \end{array} \qquad \begin{array}{cccc} & 3 & A & \\ & C & 5 & + \\ \hline & F & F & \end{array} \qquad \begin{array}{cccc} 1 & 1 & & \\ & 8 & 5 & \\ & 9 & D & + \\ \hline 1 & 2 & 2 & \end{array}$$

$$\begin{array}{cccc} 1 & 1 & & \\ & B & 9 & \\ & 7 & E & + \\ \hline 1 & 3 & 7 & \end{array} \qquad \begin{array}{cccc} 1 & 1 & & \\ & A & B & \\ & C & D & + \\ \hline 1 & 7 & 8 & \end{array} \qquad \begin{array}{cccc} 1 & 1 & & \\ & C & C & \\ & 4 & D & + \\ \hline 1 & 1 & 9 & \end{array}$$

# Binary Coded Decimal code (BCD)

In BCD code, each digit of a decimal number is represented by a 4-bit binary "code group."  Compare the binary and BCD representations of some decimal numbers.

Example 1:

For the decimal number 37,

the binary representation is 100101

the BCD representation is 0011 0111

$$\begin{array}{cc} 0011 & 0111 \\ \downarrow & \downarrow \\ 3 & 7 \end{array}$$

# Binary Coded Decimal code (BCD)

Example 2:
For the decimal number 65,
the binary representation is 1000001

the BCD representation is 0110 0101

$$\begin{array}{cc} 0110 & 0101 \\ \downarrow & \downarrow \\ 6 & 5 \end{array}$$

Example 3:
For the decimal number 123,
the binary representation is 1111011

the BCD representation is 0001 0010 0011

$$\begin{array}{ccc} 0001 & 0010 & 0011 \\ \downarrow & \downarrow & \downarrow \\ 1 & 2 & 3 \end{array}$$

# Binary Coded Decimal code (BCD)

Since there are 10 decimal digits, the 4-bit binary numbers for 0, 1, …, 9 are used for BCD representation, as shown in the table.

| decimal digit | BCD code |
|---|---|
| 0 | 0000 |
| 1 | 0001 |
| 2 | 0010 |
| 3 | 0011 |
| 4 | 0100 |
| 5 | 0101 |
| 6 | 0110 |
| 7 | 0111 |
| 8 | 1000 |
| 9 | 1001 |

# Binary Coded Decimal code (BCD)

The remaining 4-bit binary numbers 1010, 1011, 1100, 1101, 1110 and 1111, which represent A, B C, D, E and F in hexadecimal, do not correspond to any decimal digit, and are called "invalid BCD code groups".

So conversion from decimal to BCD, is simply a matter of replacing each decimal digit by the appropriate its 4-bit binary code group. Some further examples follow.

Example 5:

Consider the decimal number 478. Each of its digits is changed to its binary equivalent as shown:

| 4 | 7 | 8 |
|---|---|---|
| ↓ | ↓ | ↓ |
| 0100 | 0111 | 1000 |

# Binary Coded Decimal code (BCD)

Example 6:

Consider the decimal number 345602. Its BCD representation is as shown:

$$\begin{array}{cccccc} 3 & 4 & 5 & 6 & 0 & 2 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0011 & 0100 & 0101 & 0110 & 0000 & 0010 \end{array}$$

Conversion from BCD code to the equivalent decimal number is simply a matter of separating the BCD number into groups of 4 bits and then replacing each group by the corresponding decimal digit.

# Binary Coded Decimal code (BCD)

Example 7:

Convert the BCD number 0101011100000001 into its decimal equivalent

| 0101 | 0111 | 0000 | 0001 |
|---|---|---|---|
| ↓ | ↓ | ↓ | ↓ |
| 5 | 7 | 0 | 1 |

Example 8:

The binary sequence 0100 1011 0011 is in fact NOT a valid BCD representation, because when we try to convert it into its decimal equivalent, we find

| 0100 | 1011 | 0011 |
|---|---|---|
| ↓ | ↓ | ↓ |
| 4 | ? | 3 |

1011 is an invalid BCD code group.

# Addition of BCD numbers

In the following discussion about addition of BCD numbers it is much easier to use hexadecimal rather than binary representation.

So valid BCD code groups are represented by the hex numbers 0, 1, 2, …, 9, while the hex numbers A, B, C, D, E and F represent invalid BCD code groups.

When a microprocessor adds BCD numbers, it is really adding binary (or hex ) numbers, but the final output must look like the result of decimal addition.

Binary (or hex) additions may or may not look like decimal additions.

# Addition of BCD numbers

These three hex additions can be interpreted as correct decimal additions:

$$\begin{array}{cc}
3\ 5 & \\
2\ 4 & + \\
\hline
5\ 9 &
\end{array}
\qquad
\begin{array}{cc}
1\ 6 & \\
3\ 1 & + \\
\hline
4\ 7 &
\end{array}
\qquad
\begin{array}{cc}
4\ 3 & \\
5\ 2 & + \\
\hline
9\ 5 &
\end{array}$$

However the following three hex additions do not give correct decimal answers:

$$\begin{array}{cc}
2\ 4 & \\
3\ 6 & + \\
\hline
5\ A &
\end{array}
\qquad
\begin{array}{cc}
7\ 5 & \\
8\ 7 & + \\
\hline
F\ C &
\end{array}
\qquad
\begin{array}{cc}
3\ 8 & \\
3\ 9 & + \\
\hline
6\ 1 &
\end{array}$$

# Addition of BCD numbers

When two BCD code groups are added, there are three possible types of outcomes as illustrated by the following three cases. Remember the additions are all hex:

| (i) | (ii) | (iii) |
|---|---|---|
| $\begin{array}{r} 3\phantom{\;+} \\ 4\;+ \\ \hline 7\phantom{\;+} \end{array}$ | $\begin{array}{r} 6\phantom{\;+} \\ 7\;+ \\ \hline D\phantom{\;+} \end{array}$ | $\begin{array}{r} 9\phantom{\;+} \\ 8\;+ \\ \hline 1\;1\phantom{\;+} \end{array}$ |

In case (i), if the result is interpreted as decimal, a correct answer is obtained.

In case (ii) an invalid BCD code group is produced so the result cannot be interpreted as decimal.

In case (iii), the result, if interpreted as decimal, gives a wrong answer.

# Addition of BCD numbers

Such decimal adjustments are usually not seen by the user. In the case of a calculator, for example, the user sees only the two numbers that are input and the final output.

The hexadecimal addition of the inputs and the decimal adjustments occur inside the machine and are therefore hidden from the user. Although all the calculations are in hex (or binary), as far as the user can see the operation is simply decimal addition.

The steps involved in BCD addition are shown in the following box. Notice that the first addition and then the decimal adjustment on each column are completed before moving to the next column.

# Addition of BCD numbers

## The procedure for BCD addition

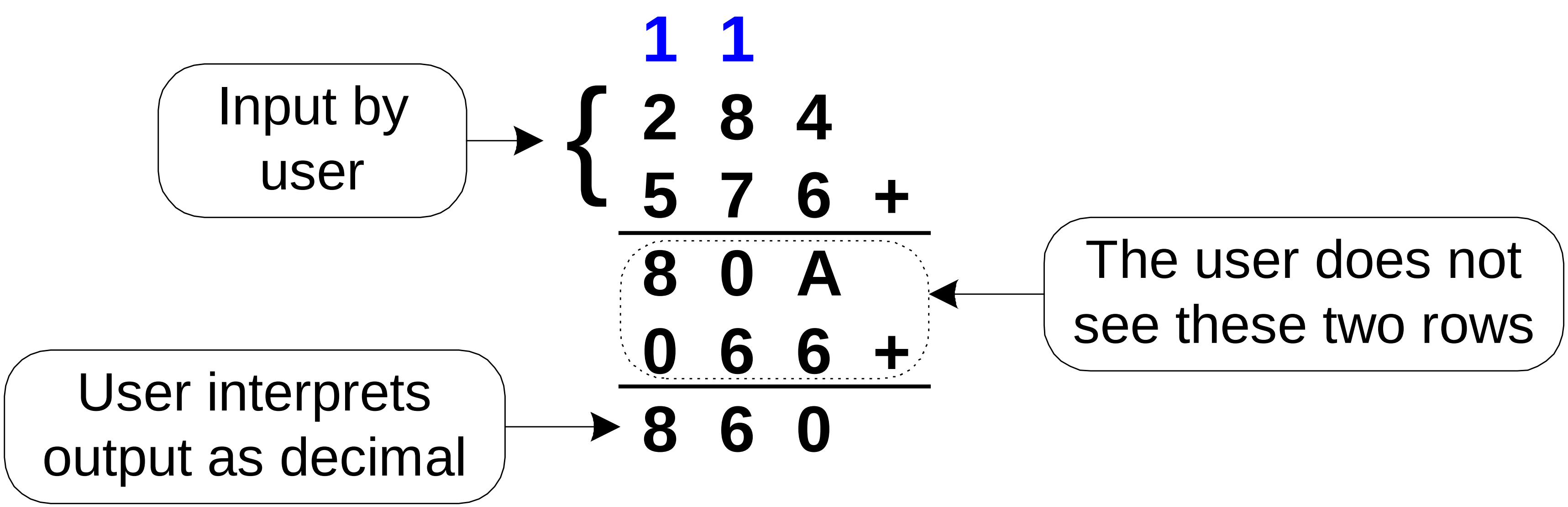
Start at the right hand column:

- add the two digits and transfer any carry to the next column;
- make the decimal adjustment by adding 6 or 0, according to the rule:
  - If the sum of the two digits is more than 9, add 6 and transfer any carry to the next column; otherwise add 0.
- Move to the next column and repeat the above operations, until all columns have been processed.

Note that a carry may come from the first addition or from the second addition.

# Addition of BCD numbers

Example 1



# Addition of BCD numbers

Example 2

Input by user

```
  1 1
  3 4 8
  2 8 9 +
  -------
  6 D 1
  0 6 6 +
  -------
  6 3 7
```

The user does not see these two rows

User interprets output as decimal

# Addition of BCD numbers

Some further examples:

| | | | |
|---|---|---|---|
| | 1 | | |
| 4 | 5 | 8 | |
| 5 | 2 | 6 | + |
| 9 | 8 | E | |
| 0 | 0 | 6 | + |
| 9 | 8 | 4 | |

| | | | |
|---|---|---|---|
| 4 | 7 | 5 | |
| 5 | 0 | 3 | + |
| 9 | 7 | 8 | |
| 0 | 0 | 0 | + |
| 9 | 7 | 8 | |

| | | | |
|---|---|---|---|
| 1 | 1 | | |
| 3 | 6 | 8 | |
| 5 | 3 | 7 | + |
| 9 | A | F | |
| 0 | 6 | 6 | + |
| 9 | 0 | 5 | |

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | | |
| | 7 | 4 | 7 | |
| | 5 | 6 | 9 | + |
| 1 | D | B | 0 | |
| 0 | 6 | 6 | 6 | + |
| 1 | 3 | 1 | 6 | |

| | | | | |
|---|---|---|---|---|
| 1 | 1 | | | |
| | 2 | 5 | 3 | |
| | 7 | 7 | 6 | + |
| 1 | A | C | 9 | |
| 0 | 6 | 6 | 0 | + |
| 1 | 0 | 2 | 9 | |

| | | | | |
|---|---|---|---|---|
| 1 | 1 | | | |
| | 9 | 8 | 2 | |
| | 6 | 4 | 7 | + |
| 1 | 0 | C | 9 | |
| 0 | 6 | 6 | 0 | + |
| 1 | 6 | 2 | 9 | |

# Addition of BCD numbers

Some further examples:

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | | |
| | 7 | 8 | 7 | 3 | |
| | 8 | 5 | 8 | 7 | + |
| 1 | 0 | E | 0 | A | |
| 0 | 6 | 6 | 6 | 6 | + |
| 1 | 6 | 4 | 6 | 0 | |

| | | | | | |
|---|---|---|---|---|---|
| 1 | | | 1 | | |
| | 3 | 1 | 0 | 9 | |
| | 7 | 0 | 8 | 7 | + |
| 1 | A | 1 | 9 | 0 | |
| 0 | 6 | 0 | 0 | 6 | + |
| 1 | 0 | 1 | 9 | 6 | |

| | | | | | |
|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | | |
| 2 | 5 | 7 | 9 | 6 | |
| | 7 | 6 | 0 | 7 | + |
| 3 | D | E | A | D | |
| 0 | 6 | 6 | 6 | 6 | + |
| 3 | 3 | 4 | 0 | 3 | |

# The End