**Department of Biomedical Engineering**
**Digital Logic Design / Second stage**
**Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi**
**Email: hussam.jawad@uomus.edu.iq**

# Lecture Two

# Arithmetic Operations and Codes

## 1- Binary Arithmetic

Binary arithmetic is essential in all digital computers and other digital systems. To understand digital systems, you must know the basics of binary addition, subtraction, multiplication, and division.

### a- Binary Addition

The four basic rules of addition binary digits are as follows:

0 + 0 = 0 Sum of 0 with a carry of 0
0 + 1 = 1 Sum of 1 with a carry of 0
1 + 0 = 1 Sum of 1 with a carry of 0
1 + 1 = 10 Sum of 0 with a carry of 1

***Example 1***

Add the following binary numbers:

*(a) 11 + 11* *(b) 100 + 10* *(c) 111 + 11* *(d) 110 + 100*

***Solution***

(a) $\begin{array}{r} 11 \\ +\,11 \\ \hline 110 \end{array}$ $\begin{array}{r} 3 \\ +\,3 \\ \hline 6 \end{array}$ **(b)** $\begin{array}{r} 100 \\ +\,10 \\ \hline 110 \end{array}$ $\begin{array}{r} 4 \\ +\,2 \\ \hline 6 \end{array}$ **(c)** $\begin{array}{r} 111 \\ +\,11 \\ \hline 1010 \end{array}$ $\begin{array}{r} 7 \\ +\,3 \\ \hline 10 \end{array}$ **(d)** $\begin{array}{r} 110 \\ +\,100 \\ \hline 1010 \end{array}$ $\begin{array}{r} 6 \\ +\,4 \\ \hline 10 \end{array}$

## b- Binary Subtraction

The four basic rules of Subtraction of binary digits are as follows:

0 - 0 = 0
1 - 1 = 0
1 - 0 = 1
10 – 1 = 1  0 – 1 with a borrow of 1

***Example (1)***

Perform the following subtraction

(a) 11 – 01  (b) 11 – 10

***Solution***

(a)
$$\begin{array}{r} 11 \\ -\ 01 \\ \hline 10 \end{array} \qquad \begin{array}{r} 3 \\ -\ 1 \\ \hline 2 \end{array}$$

(b)
$$\begin{array}{r} 11 \\ -\ 10 \\ \hline 01 \end{array} \qquad \begin{array}{r} 3 \\ -\ 2 \\ \hline 1 \end{array}$$

***Example (2)***

Subtract 011 from 101

***Solution***

$$\begin{array}{r} 101 \\ -\ 011 \\ \hline 010 \end{array} \qquad \begin{array}{r} 5 \\ -\ 3 \\ \hline 2 \end{array}$$

## c- Binary Multiplication

The four basic rules of multiplying bits are as follows:

0 × 0 = 0
0 × 1 = 0
1 × 0 = 0
1 × 1 = 1

***Example (1)***

Perform the following binary multiplications

(a) 11 × 11 (b) 101 × 111

***Solution***

(a)

```
    11      3
  × 11    × 3
  ----    ---
    11      9
  + 11
  ----
  1001
```

(b)

```
                    111      7
                  × 101    × 5
                  -----    ---
           ⎧        111     35
Partial    ⎨       000
products   ⎩    + 111
                -------
                 100011
```

**d- Binary Division**

***Example (1)***

Perform the following binary Division

(a) 110 ÷ 11 (b) 110 ÷ 10 (c) 1001011 ÷ 11

Solution

```
(a)      10     2        (b)      11     3
     11)110   3)6             10)110   2)6
         11     6                10      6
        ---    --               ---     --
        000     0                 10     0
                                  10
                                 ---
                                  00
```

(c)

```
        1 1 0 0 1
     -------------
 1 1 ) 1 0 0 1 0 1 1
     - 1 1
      ----
         1 1
       - 1 1
        -----
         0 0 0
       -     0
         -----
           0 1
        -    0
          -----
             1 1
           - 1 1
            -----
               0
```

## Complements of Binary Numbers

The 1's complement and the 2's complement of a binary number are important because they permit negative number representation. The method of 2's complement arithmetic is commonly used in computers to handle negative numbers.

## Finding the 1's Complement

The 1's complement of binary number is found by changing all 1s to 0s and all 0s to 1's as illustrated below:

10110010
↓↓↓↓↓↓↓↓
01001101

Finding the second complement

The 2's complement of a binary number is found by adding 1 to the LSB of the 1's complement.

2's complement = (1's complement) + 1

## Find the 2's complement of 10110010

***Solution***

| | |
|---|---|
| 10110010 | Binary number |
| ↓ | |
| 01001101 | 1's complement |
| + 1 | **add 1** |
| **01001110** | **2's complement** |

**Note:**

The left most bit in a signed binary number is the sign bit, which tells you whether the number is positive or negative.

A 0 sign bit indicates a positive number, and a 1 sign bit indicate a negative number.

# Hexadecimal Addition

When adding two hexadecimal numbers, use the following rules:

1- In any given column of an addition problem, think of the two hexadecimal digits in terms of their decimal value. For instance, $5_{16} = 5_{10}$ and $C_{16} = 12_{10}$.

2- If the sum of these two digits is $15_{10}$ or less, bring down the corresponding hexadecimal digit.

3- If the sum of these two digits is greater than $15_{10}$, bring down the amount of the sum that exceeds $16_{10}$ and carry a 1 to the next column.

***Example***

Add the following hexadecimal numbers:

(a) $23_{16} + 16_{16}$ (b) $58_{16} + 22_{16}$ (c) $2B_{16} + 84_{16}$ (d) $DF_{16} + AC_{16}$

***Solution***

(a) $23_{16} + 16_{16} = \mathbf{39_{16}}$

right column: $3_{16} + 6_{16} = 3_{10} + 6_{10} = 9_{10} = 9_{16}$
left column: $2_{16} + 1_{16} = 2_{10} + 1_{10} = 3_{10} = 3_{16}$

**(b)** $58_{16} + 22_{16} = \mathbf{7A_{16}}$

right column: $8_{16} + 2_{16} = 8_{10} + 2_{10} = 10_{10} = A_{16}$
left column: $5_{16} + 2_{16} = 5_{10} + 2_{10} = 7_{10} = 7_{16}$

**(c)** $2B_{16} + 84_{16} = \mathbf{AF_{16}}$

right column: $B_{16} + 4_{16} = 11_{10} + 4_{10} = 15_{10} = F_{16}$
left column: $2_{16} + 8_{16} = 2_{10} + 8_{10} = 10_{10} = A_{16}$

**(d)** $DF_{16} + AC_{16} = \mathbf{18B_{16}}$

right column: $F_{16} + C_{16} = 15_{10} + 12_{10} = 27_{10}$
$27_{10} - 16_{10} = 11_{10} = B_{16}$ with a 1 carry
left column: $D_{16} + A_{16} + 1_{16} = 13_{10} + 10_{10} + 1_{10} = 24_{10}$
$24_{10} - 16_{10} = 8_{10} = 8_{16}$ with a 1 carry

## Hexadecimal Subtraction

There are several methods to subtract hexadecimal, one of the most common methods is converting the hexadecimal number to binary. Take the 2's complement of the binary number. Convert the result to hexadecimal.

***Example***

Subtract the following hexadecimal numbers:

(a) $84_{16} - 2A_{16}$  (b) $C3_{16} - 0B_{16}$

***Solution***

(a)

2A = 00101010 Binary number

↓

11010101 1's complement
+ 1 **add 1**
**11010110 = D6 2's complement**

**84**
**+ D6**
**~~1~~5A** Drop carry

The difference is $5A_{16}$

(b)

0B = 00001011 Binary number

11110100 1's complement
+ **1** **add 1**
**11110101 = F5 2's complement**

**C3**
**+ F5** **add**
**~~1~~B8** Drop carry

The difference is $B8_{16}$

## Binary Coded Decimal (BCD)

Binary coded decimal (BCD) is a way to express each of the decimal digits with a binary code. There are only ten code group in the BCD system, so it is very easy to convert between decimal and BCD. Because we like to read and write in decimal, the BCD code provides an excellent interface to a binary system.

### The 8421 BCD code

The 8421 code is a type of BCD code. Binary coded decimal means that each decimal digit, 0 through 9, is represented by a binary code of four bits. The designation 8421 indicates the binary weight of the four bits ($2^3$, $2^2$, $2^1$, $2^0$)

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

### Invalid codes

In the 8421 code, only ten of (0000 through 1111) are used. The six code combinations that are not used are 1010, 1011, 1100, 1101, 1110, and 1111 which are invalid in the 8421 BCD code.

***Example:*** Convert each of the following decimal numbers to BCD

(a) 35 (b) 98 (c) 170 (d) 2469

***Solution***

(a) 3 5 → **00110101**

(b) 9 8 → **10011000**

(c) 1 7 0 → **000101110000**

(d) 2 4 6 9 → **0010010001101001**

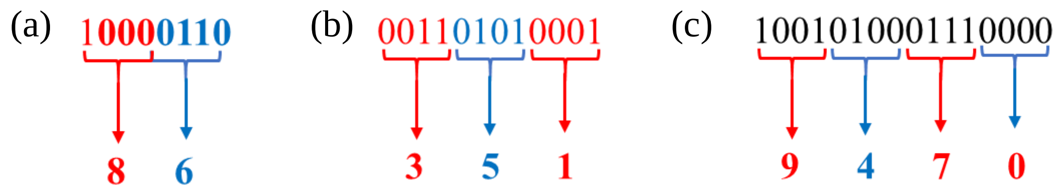
**Department of Biomedical Engineering**
**Digital Logic Design / Second stage**
**Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi**
**Email: hussam.jawad@uomus.edu.iq**

***Example 2***

Convert each of the following BCD codes to decimal

(a) 10000110 (b) 001101010001 (c) 1001010001110000

***Solution***




***BCD addition***

BCD is a numerical code and can be used in arithmetic operations.
The steps of adding two BCD numbers:

1- Add the two BCD numbers using the role of binary.
2- If a 4-bit sum is equal to or less than 9, it is a valid BCD number.
3- If a 4-bit sum is greater than 9, or if a carry-out of the 4-bit group is generated, it is an invalid result. Add 6 (0110) to the 4-bit sum to skip the six invalid states and return the code to 8421.

***Example1***

Add the following BCD numbers:

(a) 0011 + 0100 (b) 00100011 + 00010101

(c) 10000110 + 00010011 (d) 01001010000 + 010000010111

**Solution**

(a)
```
  0011        3
+ 0100      + 4
  0111        7
```

(b)
```
  0010 0011     23
+ 0001 0101   + 15
  0011 1000     38
```

(c)
```
  1000 0110     86
+ 0001 0011   + 13
  1001 1001     99
```

(d)
```
  0100 0101 0000     450
+ 0100 0001 0111   + 417
  1000 0110 0111     867
```

***Example:***

Add the following BCD numbers:

**(a)** 1001 + 0100 **(b)** 1001 + 1001

**(c)** 00010110 + 00010101 **(d)** 01100111 + 01010011

## Solution

The decimal number additions are shown for comparison.

```
(a)              1001                                           9
               + 0100                                          +4
                 1101     Invalid BCD number (>9)              13
               + 0110     Add 6
        0001     0011     Valid BCD number
          ↓        ↓
          1        3
```

```
(b)              1001                                           9
               + 1001                                         + 9
             1   0010     Invalid because of carry             18
               + 0110     Add 6
        0001     1000     Valid BCD number
          ↓        ↓
          1        8
```

```
(c)     0001     0110                                          16
      + 0001     0101                                        + 15
        0010     1011     Right group is invalid (>9),         31
                             left group is valid.
               + 0110     Add 6 to invalid code. Add
                             carry, 0001, to next group.
        0011     0001     Valid BCD number
          ↓        ↓
          3        1
```

```
(d)              0110        0111                                        67
               + 0101        0011                                      + 53
                 1011        1010     Both groups are invalid (>9)      120
               + 0110      + 0110     Add 6 to both groups
        0001     0010        0000     Valid BCD number
          ↓        ↓           ↓
          1        2           0
```

# Gray code

## Binary to Gray code conversion

There are two steps in conversion from binary to gray code

1- The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary.

2- Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard the carries.

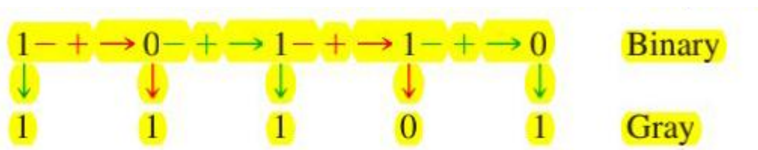
***Example 1***

(a) Convert the binary number 10110 to Gray code

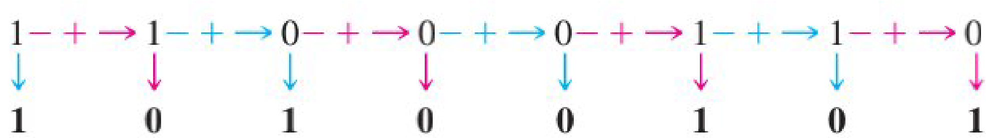
(b) Convert the binary number 11000110 to Gray code

***Solution***

(a)




(b)



**Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi**
**Email: hussam.jawad@uomus.edu.iq**

## Gray to binary code conversion

There are two steps in conversion from gray code to binary

1- The most significant bit (left-most) in the binary code is the same as the corresponding bit in the Gray code.

2- Add each binary code bit generated to the Gray code bit in the next adjacent position. Discard carries.

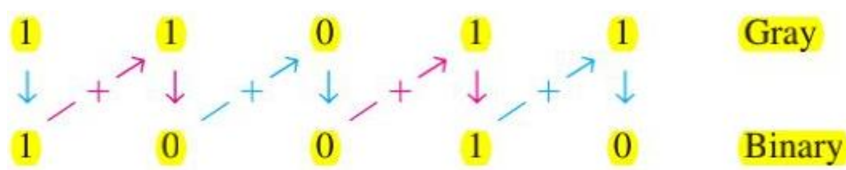
***Example***

(a) Convert the Gray code 11011 to binary

(b) Convert the Gray code 10101111 to binary

***Solution***

(a)




(b)