CME 2003
Digital Logic

Operations & Codes

Şerife SUNGUN

Addition of Signed Numbers

- Both numbers positive
- Positive number with magnitude larger than negative number
- Negative number with magnitude larger than positive number
- Both numbers negative

... Addition of Signed Numbers

- Both numbers positive:

```
  00000111
+ 00000100
  --------
  00001011
```

Addition of two positive numbers yields a positive number.

... Addition of Signed Numbers

- Positive number with magnitude larger than negative number:

```
   00001111
 + 11111010
 ----------
  100001001
```

Discard carry

Addition of a positive number and a smaller negative number yields a positive number.

... Addition of Signed Numbers

- Negative number with magnitude larger than positive number:

```
  00010000
+ 11101000
  --------
  11111000
```

Addition of a positive number and a larger negative number yields a negative number in 2's complement.

... Addition of Signed Numbers

- Both numhers negative:

```
   11111011
 + 11110111
 ----------
 1 11110010
```

Discard carry

Addition of two negative numbers yields a <u>negative</u> number in 2's complement.

Overflow Condition

- An overflow results as indicated by an incorrect sign bit when two positive or two negative numbers are added.

```
    01111101        11011101
  + 00111010      + 10001101
  ----------      ----------
    10110111        01101010
```

overflow

overflow

Sign incorrect

Magnitude incorrect

Subtraction

- To subtract two signed numbers,
  - Take the 2's complement of the subtrahend
  - Add.
  - Discard any final carry bit.

8 - 3 = 8 + (-3) = 5.

```
  00001000            00001000
 -11111101           +11111101
                     ~~1~~00000101
```

Discard carry

Multiplication

Methods of Multiplication :

- Direct addition method.
- Partial products method

$$\begin{array}{rl} 8 & \text{Multiplicand} \\ \times\ 3 & \text{Multiplier} \\ \hline 24 & \text{Product} \end{array}$$

Direct Addition Method

Multiplication is equivalent to adding a number to itself a number of times equal to the multiplier.

Example

Multiply the signed binary numbers: 01001101 (multiplicand) and 00000100 (multiplier) using the direct addition method.

01001101	1st time
+ 01001101	2nd time
10011010	Partial sum
+ 01001101	3rd time
11100111	Partial sum
+ 01001101	4th time
100110100	Product

The Partial Products Method

- The multiplicand is multiplied by **each multiplier digit** beginning with the least significant digit.
- The result of the multiplication of the multiplicand by a multiplier digit is called a **partial product**.
- Each successive partial product is moved (shifted) one place to the left and when all the partial products have been produced, they are added to get the final product.

... The Partial Products Method

- **Example**

239	Multiplicand
$\times$ 123	Multiplier
717	1st partial product (3×239)
478	2nd partial product (2×239)
+ 239	3rd partial product (1×239)
29,397	Final product

The Basic Steps in the Partial Products Method

- **Step 1:** Determine if the signs of the multiplicand and multiplier are the same or different. This determines what the sign of the product will be.
  - If the signs are the same, the product is **positive**.
  - If the signs are **different**, the product is **negative**.

... The Basic Steps in the Partial Products Method

- **Step 2:** Change any negative number **to true (uncomplemented) form**. Because most computers store negative numbers in 2's complement, a 2's complement operation is required to get the negative number into true form.

- **Step 3:** Starting with the least significant multiplier bit, **generate the partial products**. When the multiplier bit is 1, the partial product is the same as the multiplicand. When the multiplier bit is 0, the partial product is zero. Shift each successive partial product one bit to the left.

... The Basic Steps in the Partial Products Method

- **Step 4:** Add each successive partial product to the sum of the previous partial products to get the final product.

- **Step 5:** If the sign bit that was determined in step 1 is **negative**, **take the 2's complement** of the product. If **positive**, **leave the product in true form**. Attach the sign bit to the product.

Example

01010011 x 11000101 = ?

x 00111011

2's complement

1010011	Multiplicand
× 0111011	Multiplier
1010011	1st partial product
+ 1010011	2nd partial product
11111001	Sum of 1st and 2nd
+ 0000000	3rd partial product
011111001	Sum
+ 1010011	4th partial product
1110010001	Sum
+ 1010011	5th partial product
100011000001	Sum
+ 1010011	6th partial product
1001100100001	Sum
+ 0000000	7th partial product
1001100100001	Final product

1001100100001 ⟶ 0110011011111

Attach the sign bit ⟶ **1 0110011011111**

Division

- Standard division format: $$\frac{\text{dividend}}{\text{divisor}} = \text{quotient}$$

- The division operation in computers is accomplished using **subtraction**.

21	Dividend
− 7	1st subtraction of divisor
14	1st partial remainder
− 7	2nd subtraction of divisor
7	2nd partial remainder
− 7	3rd subtraction of divisor
0	Zero remainder

The basic steps in a division process

- **Step 1.** Determine if the **signs** of the dividend and divisor are the **same or different**. This determines what the sign of the quotient will be. Initialize the quotient to zero.
  - If the signs are the same, the quotient is positive.
  - If the signs are different, the quotient is negative.

- **Step 2.** Subtract the divisor from the dividend **using 2's complement addition** to get the first partial remainder and add 1 to the quotient. If this partial remainder is positive, go to step 3. If the partial remainder is zero or negative, the division is complete.

- **Step 3.** Subtract the **divisor from the partial remainder** and add 1 to the quotient. If the result is positive, repeat for the next partial remainder. If the result is zero or negative, the division is complete.

Example

Divide 01100100 by 00011001.

Step 1: The signs of both numbers are positive, so the quotient will be positive. The quotient is initially zero: 00000000.

Step 2: Subtract the divisor from the dividend using 2's complement addition (remember that final carries are discarded).

```
   01100100  Dividend
 + 11100111  2's complement of divisor
 ----------
   01001011  Positive 1st partial remainder
```

Add 1 to quotient: 00000000 + 00000001 = 00000001

… Example

Step 3: Subtract the divisor from the 1st partial remainder using 2's complement addition.

```
    01001011  1st partial remainder
+   11100111  2's complement of divisor
------------
    00110010  Positive 2nd partial remainder
```

… Example

Step 4: Subtract the divisor from the 2nd partial remainder using 2's complement addition.

```
  00110010  2nd partial remainder
+ 11100111  2's complement of divisor
  --------
  00011001  Positive 3rd partial remainder
```

Add 1 to quotient:

00000010 + 00000001 = 00000011.

… Example

Step 5: Subtract the divisor from the 3rd partial remainder using 2's complement addition.

```
   00011001  3rd partial remainder
 + 11100111  2's complement of divisor
 ----------
   00000000  Zero remainder
```

Add 1 to quotient:

00000011 + 00000001 = 00000100 (final quotient).

The process is complete.

Hexadecimal Numbers

The hexadecimal number system has

- A base of 16
- consists of digits 0-9 and letters A-F.

... Hexadecimal Numbers

DECIMAL	BINARY	HEXADECIMAL
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

Binary-to-Hexadecimal Conversion

- break the binary number into 4-bit groups
- starting at the right-most bit
- replace each 4-bit group with the equivalent hexadecimal

Example

Convert the following binary numbers to hexadecimal:

(a) 100101001010111 (b) 111111000101101001

$$\underbrace{1100}_{C}\,\underbrace{1010}_{A}\,\underbrace{0101}_{5}\,\underbrace{0111}_{7} = CA57_{16}$$

$$\underbrace{0011}_{3}\,\underbrace{1111}_{F}\,\underbrace{0001}_{1}\,\underbrace{0110}_{6}\,\underbrace{1001}_{9} = 3F169_{16}$$

Hexadecimal-to-Binary Conversion

Replace each hexadecimal symbol with the appropriate four bits.

Example

Determine the binary numbers for the following hexadecimal numbers:

(a) 10A4 (b) CF8E (c) 9742

(a) 1 0 A 4 → 1000010100100

(b) C F 8 E → 1100111110001110

(c) 9 7 4 2 → 1001011101000010

Hexadecimal-to-Decimal Conversion

- Multiply the decimal value of each hexadecimal digit by its weight and take the sum of these products.
- The weights of a hexadecimal number are increasing powers of 16 (from right to left).

16^3	16^2	16^1	16^0
4096	256	16	1

Example $E516 = (?)_{10}$

$$\begin{aligned} E516 &= (E \times 16) + (5 \times 1) = (14 \times 16) + (5 \times 1) \\ &= 224 + 5 \\ &= (229)_{10} \end{aligned}$$

Decimal-to-Hexadecimal Conversion

- Repeated division of a decimal number by 16 will produce the equivalent hexadecimal number, formed by the remainders of the divisions.
- The first remainder produced is the least significant digit
- When a quotient has a fractional part, the fractional part is multiplied by the divisor to get the remainder.

Example

Convert the decimal number 650 to hexadecimal.

$$\frac{650}{16} = 40.625 \rightarrow 0.625 \times 16 = 10 = \text{A}$$

$$\frac{40}{16} = 2.5 \rightarrow 0.5 \times 16 = 8 = 8$$

$$\frac{2}{16} = 0.125 \rightarrow 0.125 \times 16 = 2 = 2$$

Stop when whole number quotient is zero.

2 8 A Hexadecimal number

MSD LSD

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 values.

   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.

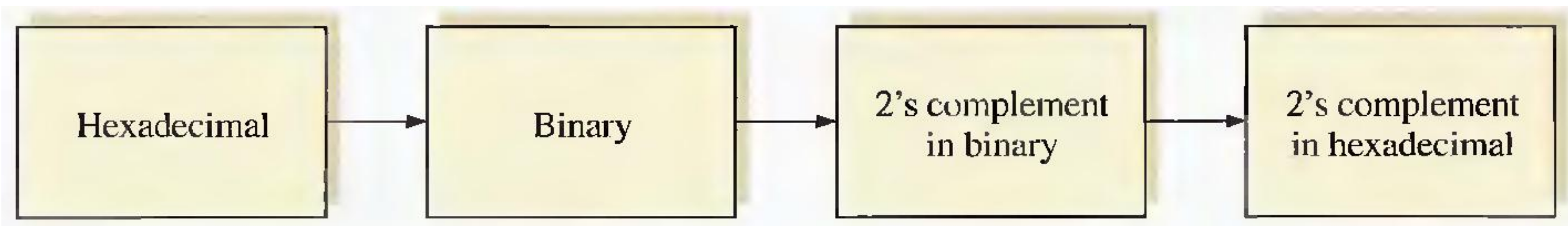
Example

Add the following hexadecimal numbers:

$$\begin{array}{r} 23_{16} \\ +16_{16} \\ \hline 39_{16} \end{array} \qquad \begin{array}{r} 58_{16} \\ +\ 22_{16} \\ \hline 7A_{16} \end{array} \qquad \begin{array}{r} 2B_{16} \\ +\ 84_{16} \\ \hline AF_{16} \end{array} \qquad \begin{array}{r} DF_{16} \\ +\ AC_{16} \\ \hline 18B_{16} \end{array}$$

Hexadecimal Subtraction - Method 1

- Convert the hexadecimal number to binary.
- Take the 2's complement of the binary number.
- Convert the result to hexadecimal.

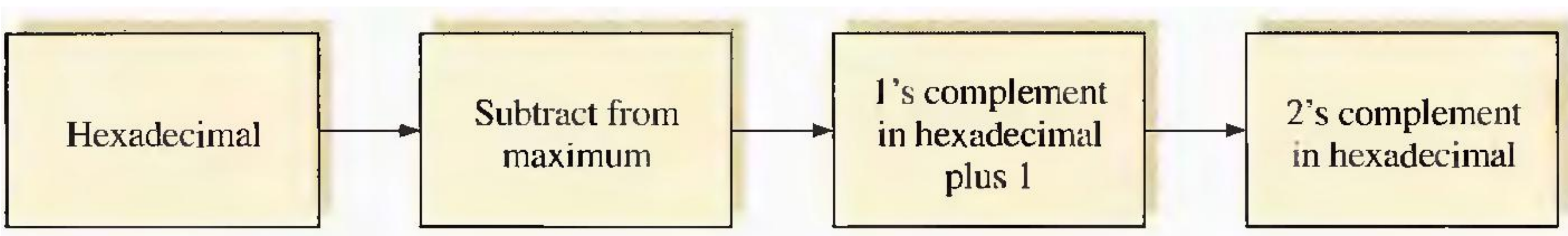


Example

2A → 00101010 → 11010110 → D6

Hexadecimal Subtraction - Method 2

- Subtract the hexadecimal number from the maximum hexadecimal number and add .

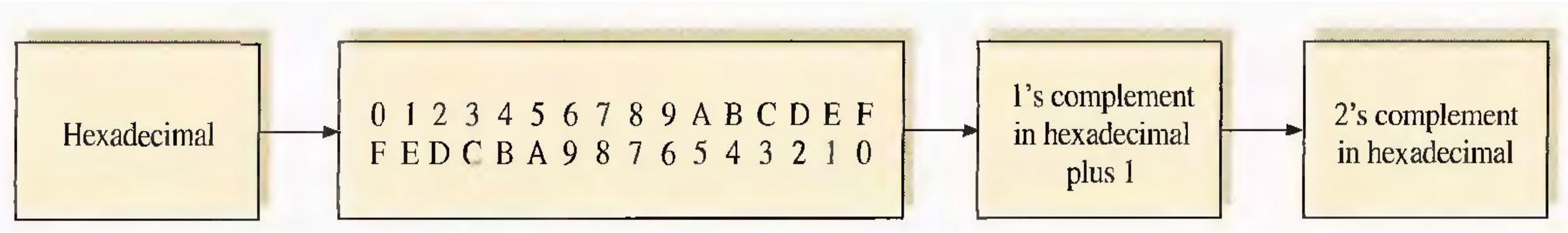


Example

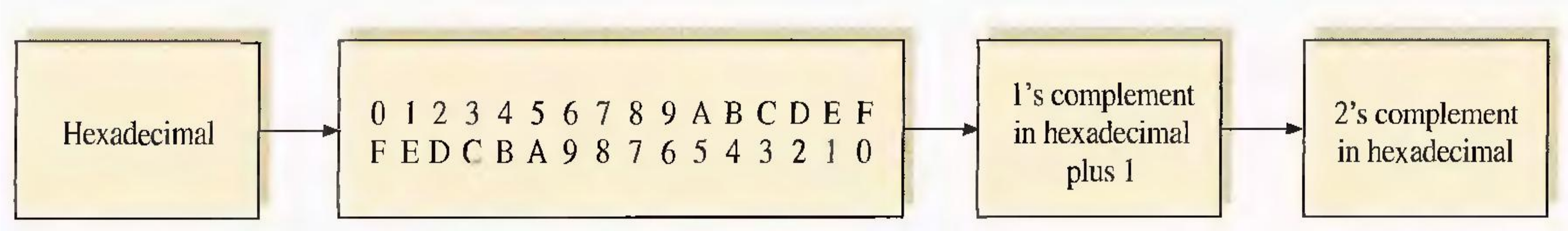
2A → FF – 2A → D5 + 1 → D6

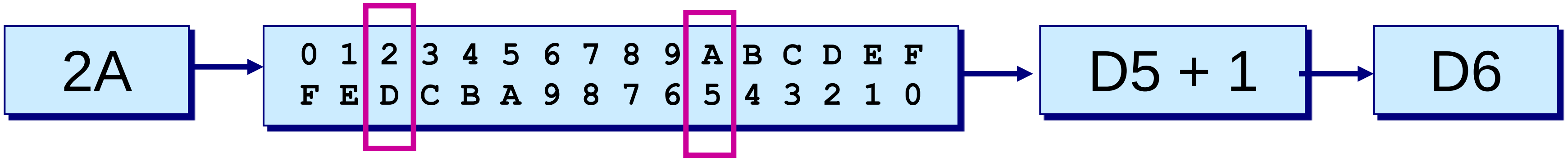
Hexadecimal Subtraction - Method 3

- Write the sequence of single hexadecimal digits.
- Write the sequence in reverse below the forward sequence.
- The 1's complement of each hex digit is the digit directly below it.
- Add 1 to the resulting number to get the 2's complement.



Example





$84_{16} - 2A_{16} = ?$

$$\begin{array}{r} 84 \\ -\ 2A \\ \hline \end{array} \qquad \begin{array}{r} 84 \\ +D6 \\ \hline \cancel{1}\,5A \end{array}$$

Drop carry

Result : 5A

Example

$C3_{16}$ - $0B_{16}$ = ?

$0B_{16}$ = 00001011

2's complement of $0B_{16}$ = 11110101

= $F5_{16}$ (using Method 1)

```
   C3
+  F5   Add
------
  1B6   Drop carry
```

The difference is B6.

Octal Numbers

The octal number system is composed of eight digits:

0, 1, 2, 3, 4, 5, 6, 7

Octal-to-Decimal Conversion

Each successive digit position is an increasing power of eight, beginning in the right-most column with 8°.

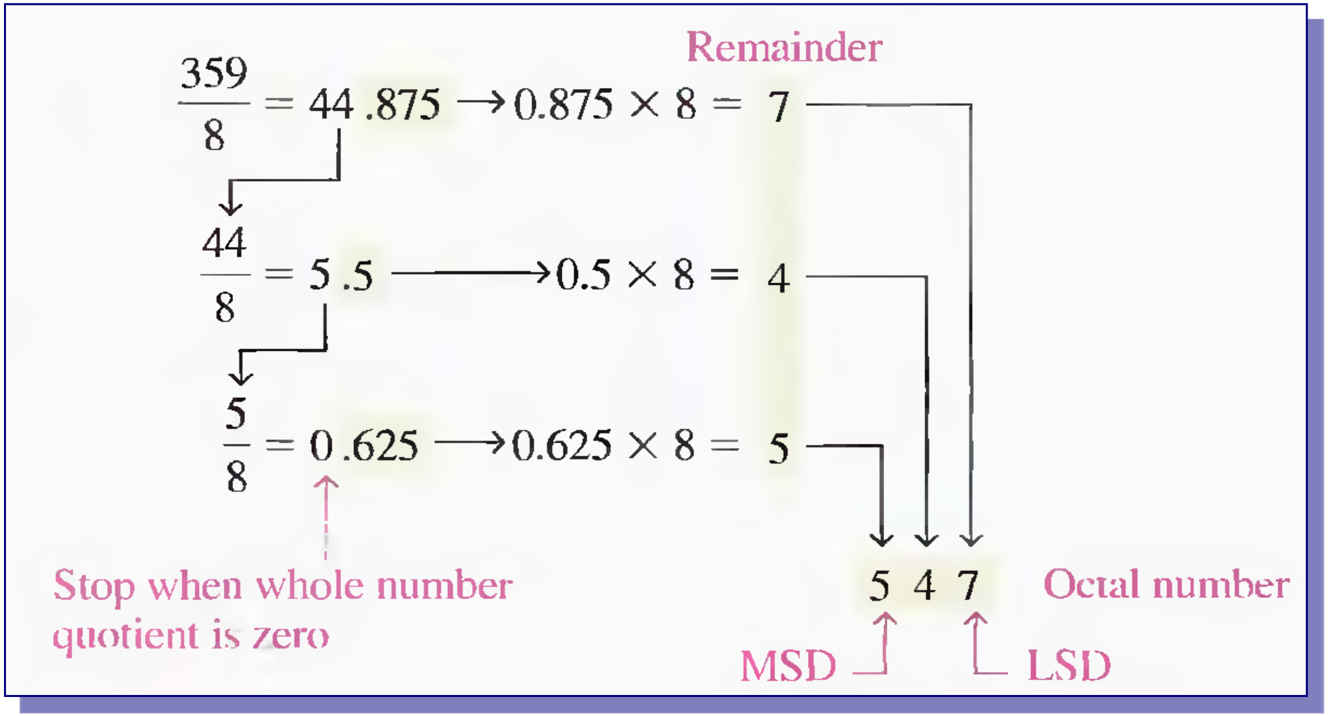
Example

	8^3	8^2	8^1	8^0
Weight:	8^3	8^2	8^1	8^0
Octal number:	2	3	7	4

$$
\begin{aligned}
2374_8 &= (2 \times 8^3) + (3 \times 8^2) + (7 \times 8^1) + (4 \times 8^0) \\
&= (2 \times 512) + (3 \times 64) + (7 \times 8) + (4 \times 1) \\
&= 1024 + 192 + 56 + 4 = 1276_{10}
\end{aligned}
$$

Decimal-to-Octal Conversion

- Each successive division by 8 yields a remainder that becomes a digit in the equivalent octal number.
- The first remainder generated is the least significant digit (LSD).



Octal-to-Binary Conversion

Because each octal digit can be represented by a 3-bit binary number

OCTAL DIGIT	0	1	2	3	4	5	6	7
BINARY	000	001	010	011	100	101	110	111

Example Convert each of the following octal numbers to binary:

(a) 13_8 (b) 25_8 (c) 140_8 (d) 7526_8

(a) 1 3 → $\underbrace{001}\underbrace{011}$ = 001011

(b) 2 5 → $\underbrace{010}\underbrace{101}$ = 010101

(c) 1 4 0 → $\underbrace{001}\underbrace{100}\underbrace{000}$ = 001100000

(d) 7 5 2 6 → $\underbrace{111}\underbrace{101}\underbrace{010}\underbrace{110}$ = 111101010110

Binary-to-Octal Conversion

Example

Convert each of the following binary numbers to octal:

(a) 1101O1 (b) 101111O01 (c) 100110011010

$$\text{(a)}\quad \underbrace{110}_{6}\,\underbrace{101}_{5} = \mathbf{65_8}$$

$$\text{(b)}\quad \underbrace{101}_{5}\,\underbrace{111}_{7}\,\underbrace{001}_{1} = \mathbf{571_8}$$

$$\text{(c)}\quad \underbrace{100}_{4}\,\underbrace{110}_{6}\,\underbrace{011}_{3}\,\underbrace{010}_{2} = \mathbf{4632_8}$$

Binary Coded Decimal

- The 8421 Code
  - each decimal digit, 0 through 9, is represented by a binary code of four bits.
  - The designation 8421 indicates the binary weights 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

… The 8421 Code

Invalid Codes

- Sixteen numbers (0000 through 1111) can be represented with 4 bits but in the 8421 code only ten of these are used.
- The invalid six code :

  1010, 1011, 1100, 1101, 1110, and 1111

Example

- Convert each of the following decimal numbers to BCD:

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

(a) $\overbrace{0011}^{3}\overbrace{0101}^{5}$

(b) $\overbrace{1001}^{9}\overbrace{1000}^{8}$

(c) $\overbrace{0001}^{1}\overbrace{0111}^{7}\overbrace{0000}^{0}$

(d) $\overbrace{0010}^{2}\overbrace{0100}^{4}\overbrace{0110}^{6}\overbrace{1001}^{9}$

Example

Convert each of the following BCD codes to decimal:

(a) 10000110

86

(b) 001101010001

351

(c) 1001010001110000

9470

BCD Addition

- **Step 1** Add the two BCD numbers, using the rules for binary addition.
- **Step 2** If a 4-bit sum is equal to or less than 9, it is a valid BCD number.
- **Step 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 in order to skip the six invalid states.
  - If a carry results when 6 is added. simply add the carry to the next 4-bit group.

Example

Addition of the BCD numbers

$$\begin{array}{rrl}
 & 1001 & \\
 & +\ 0100 & \\
\hline
 & 1101 & \text{Invalid BCD number } (>9) \\
 & +\ 0110 & \text{Add 6} \\
\hline
\underbrace{0001}_{\downarrow\ 1} & \underbrace{0011}_{\downarrow\ 3} & \text{Valid BCD number}
\end{array}$$

Example

- Addition of the BCD numbers

$$
\begin{array}{rrrl}
 & 0110 & 0111 & \\
 & +\,0101 & 0011 & \\
\cline{2-3}
 & 1011 & 1010 & \text{Both groups are invalid (>9)} \\
 & +\,0110 & +\,0110 & \text{Add 6 to both groups} \\
\hline
\underbrace{0001}_{\substack{\downarrow \\ 1}} & \underbrace{0010}_{\substack{\downarrow \\ 2}} & \underbrace{0000}_{\substack{\downarrow \\ 0}} & \text{Valid BCD number}
\end{array}
$$

Digital Codes

- The Gray Code
  - The Gray code is unweighted and is not an arithmetic code
  - weights assigned to the bit positions.
  - only a single bit change from one code word to the next in sequence.
  - the Gray code can have any number of bits.

… Gray Code

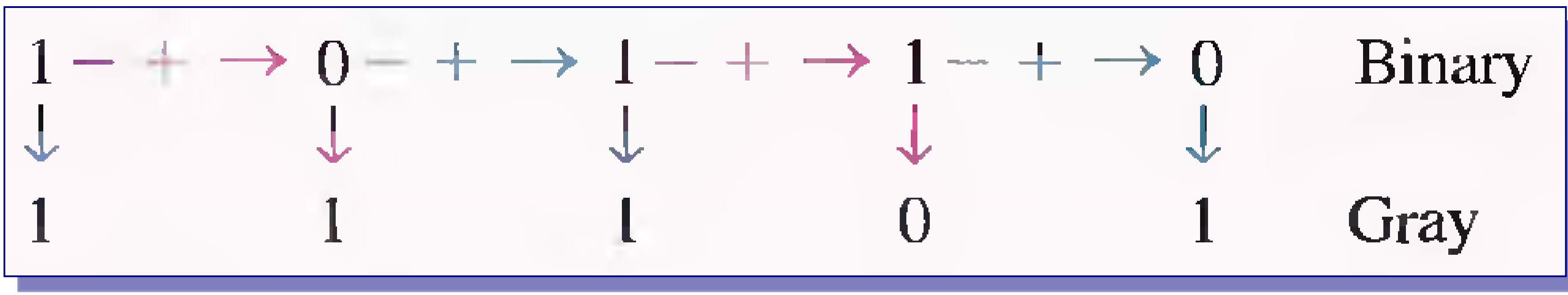
DECIMAL	BINARY	GRAY CODE	DECIMAL	BINARY	GRAY CODE
0	0000	0000	8	1000	1100
1	0001	0001	9	1001	1101
2	0010	0011	10	1010	1111
3	0011	0010	11	1011	1110
4	0100	0110	12	1100	1010
5	0101	0111	13	1101	1011
6	0110	0101	14	1110	1001
7	0111	0100	15	1111	1000

Binary-to-Gray Code Conversion

1. The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary number.
2. Going from left to right, add each adjacent pair of binary code bits to get the next Gray code bit. Discard carries.

Example

The conversion of the binary number 10110 to Gray code :



The Gray code is 11101

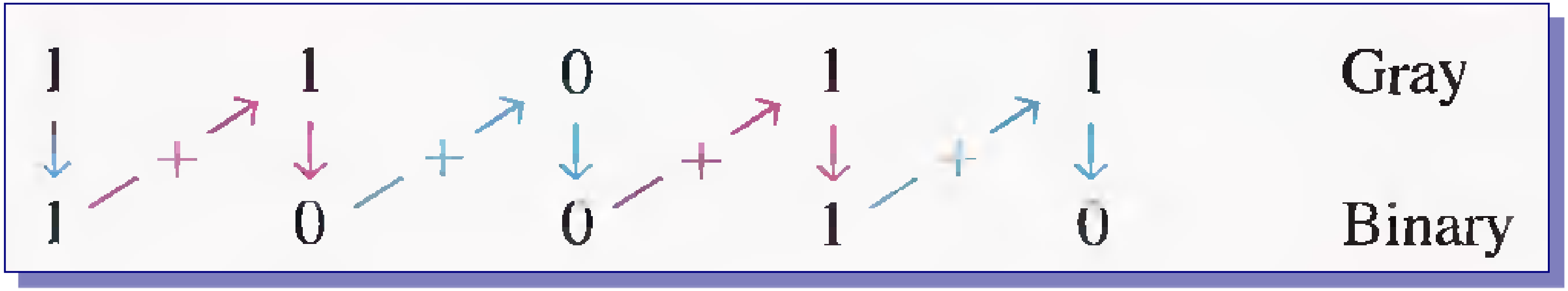
Gray-to-Binary Conversion

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.

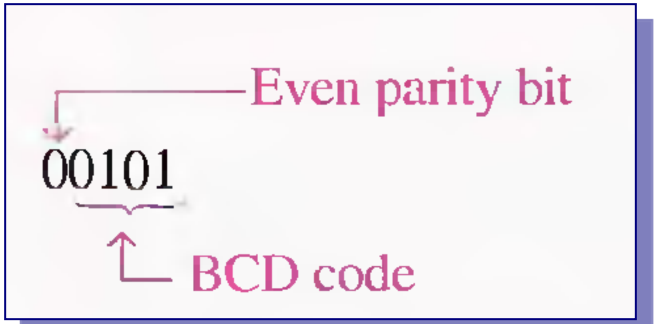
Example

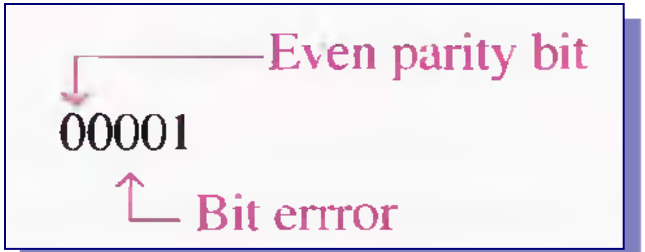
The conversion of the Gray code word 11011 to binary



Error Detection and Correction Codes

Parity Method





EVEN PARITY		ODD PARITY	
P	BCD	*P*	BCD
0	0000	1	0000
1	0001	0	0001
1	0010	0	0010
0	0011	1	0011
1	0100	0	0100
0	0101	1	0101
0	0110	1	0110
1	0111	0	0111
1	1000	0	1000
0	1001	1	1001