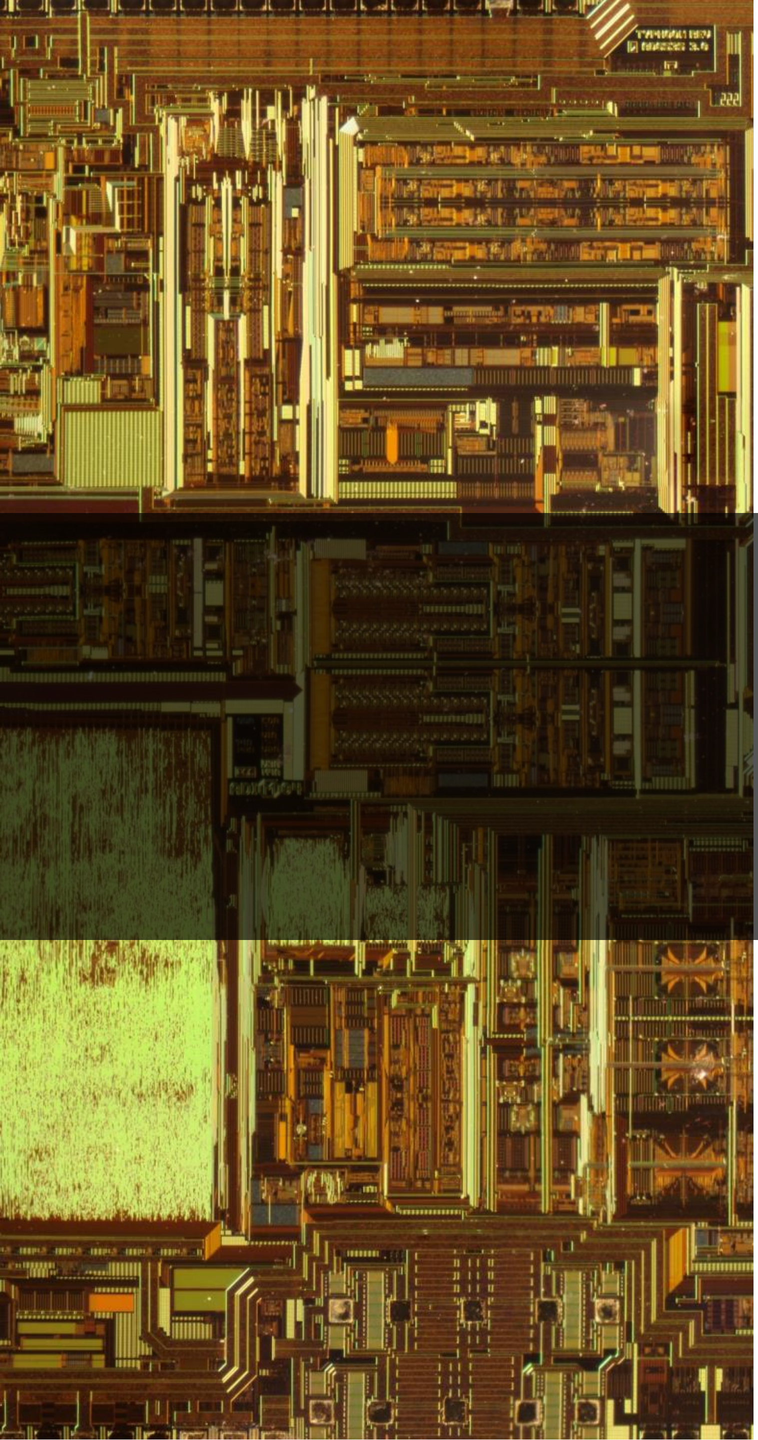


University of Al Maarif

Department of Medical Instruments Techniques Engineering

Class: 2nd

Digital Electronics

MIET2203

Lecture 4: Arithmetic Operation

Lecturer: Mohammed Jabal

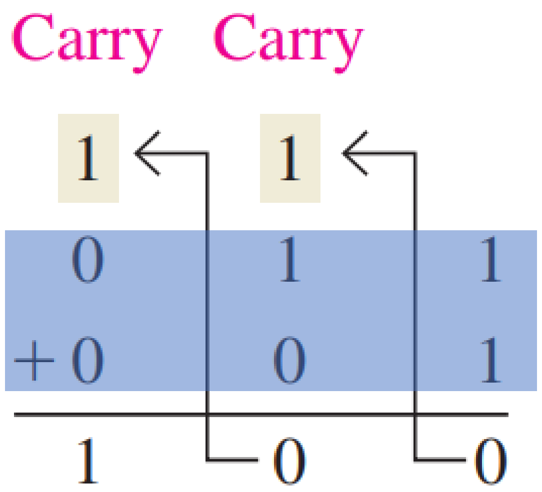
2024 - 2025

1. Binary Addition

- **The four basic rules for adding binary digits (bits) 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

In binary $1 + 1 = 10$, not 2.



1. Binary Addition

EXAMPLE 2–7

Add the following binary numbers:

(a) 11 + 11 **(b)** 100 + 10

(c) 111 + 11 **(d)** 110 + 100

Solution

The equivalent decimal addition is also shown for reference.

(a)
$$\begin{array}{r} 11 \\ +\ 11 \\ \hline \mathbf{110} \end{array} \qquad \begin{array}{r} 3 \\ +\ 3 \\ \hline 6 \end{array}$$

(b)
$$\begin{array}{r} 100 \\ +\ 10 \\ \hline \mathbf{110} \end{array} \qquad \begin{array}{r} 4 \\ +\ 2 \\ \hline 6 \end{array}$$

(c)
$$\begin{array}{r} 111 \\ +\ 11 \\ \hline \mathbf{1010} \end{array} \qquad \begin{array}{r} 7 \\ +\ 3 \\ \hline 10 \end{array}$$

(d)
$$\begin{array}{r} 110 \\ +\ 100 \\ \hline \mathbf{1010} \end{array} \qquad \begin{array}{r} 6 \\ +\ 4 \\ \hline 10 \end{array}$$

Related Problem

Add 1111 and 1100.

2. Binary Subtraction

- **The four basic rules for subtracting bits are as follows:**

$$0 - 0 = 0$$
$$1 - 1 = 0$$
$$1 - 0 = 1$$
$$10 - 1 = 1 \qquad 0 - 1 \text{ with a borrow of } 1$$

EXAMPLE 2–8

Perform the following binary subtractions:

(a) 11 − 01 **(b)** 11 − 10

Solution

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

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

No borrows were required in this example. The binary number 01 is the same as 1.

2. Binary Subtraction

EXAMPLE 2–9

Subtract 011 from 101.

Solution

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

Let's examine exactly what was done to subtract the two binary numbers since a borrow is required. Begin with the right column.

Left column:
When a 1 is borrowed,
a 0 is left, so $0 - 0 = 0$.

Middle column:
Borrow 1 from next column
to the left, making a 10 in
this column, then $10 - 1 = 1$.

$$\begin{array}{r} \overset{0}{\cancel{1}}{}^{1}01 \\ -0\,11 \\ \hline 0\,10 \end{array}$$

Right column:
$1 - 1 = 0$

3. Binary Multiplication

- **The four basic rules for multiplying bits are as follows:**

$$0 \times 0 = 0$$
$$0 \times 1 = 0$$
$$1 \times 0 = 0$$
$$1 \times 1 = 1$$

Binary multiplication of two bits is the same as multiplication of the decimal digits 0 and 1.

EXAMPLE 2–10

Perform the following binary multiplications:

(a) 11×11 **(b)** 101×111

Solution

(a)

$$\begin{array}{r} 11 \\ \times\ 11 \\ \hline 11 \\ +11 \\ \hline \mathbf{1001} \end{array} \quad \text{Partial products} \qquad \begin{array}{r} 3 \\ \times\ 3 \\ \hline 9 \end{array}$$

(b)

$$\begin{array}{r} 111 \\ \times\ 101 \\ \hline 111 \\ 000 \\ +111 \\ \hline \mathbf{100011} \end{array} \quad \text{Partial products} \qquad \begin{array}{r} 7 \\ \times\ 5 \\ \hline 35 \end{array}$$

4. Binary Division

- **Division in binary follows the same procedure as division in decimal, as Example 2–11 illustrates. The equivalent decimal divisions are also given.**

EXAMPLE 2–11

Perform the following binary divisions:

(a) 110 ÷ 11 **(b)** 110 ÷ 10

Solution

(a)
$$\begin{array}{r} \mathbf{10} \\ 11\overline{)110} \\ \underline{11} \\ 000 \end{array} \qquad \begin{array}{r} 2 \\ 3\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

(b)
$$\begin{array}{r} \mathbf{11} \\ 10\overline{)110} \\ \underline{10} \\ 10 \\ \underline{10} \\ 00 \end{array} \qquad \begin{array}{r} 3 \\ 2\overline{)6} \\ \underline{6} \\ 0 \end{array}$$

Example:-

1001 (dividend)
÷ 011 (divisor)

$$\begin{array}{r} 0\ 0\ 1\ 1 \\ 0\ 1\ 1\overline{)1\ 0\ 0\ 1} \\ \underline{0\ 1\ 1\ \ } \\ 0\ 0\ 1\ 1 \\ \underline{0\ 0\ 1\ 1} \\ 0\ 0\ 0\ 0 \end{array}$$

Complements of Binary Numbers

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

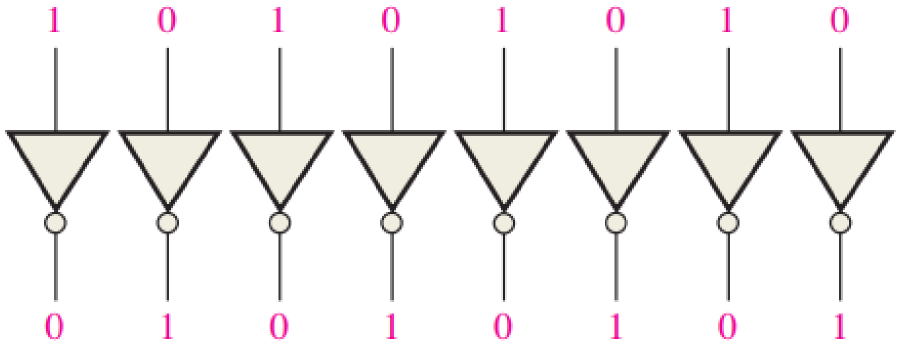
Complements of Binary Numbers

Finding the 1's Complement

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

1 0 1 1 0 0 1 0	Binary number
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓	
0 1 0 0 1 1 0 1	1's complement

The simplest way to obtain the 1's complement of a binary number with a digital circuit is to use parallel inverters (NOT circuits), as shown in Figure 2–2 for an 8-bit binary number.



Complements of Binary Numbers

Finding the 2's Complement

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

$$\text{2's complement} = (\text{1's complement}) + 1$$

EXAMPLE 2–12

Find the 2's complement of 10110010.

Solution

10110010	Binary number
01001101	1's complement
+ 1	Add 1
01001110	2's complement

Related Problem

Determine the 2's complement of 11001011.

Signed Numbers

- Digital systems, such as the computer, must be able to handle both **positive** and **negative** numbers. A **signed** binary number consists of both **sign** and **magnitude** information. The **sign** indicates whether a number is **positive** or **negative**, and the **magnitude** is the **value** of the number.

The Sign Bit

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 indicates a negative number.

Signed Numbers

Sign-Magnitude Form

+25 is expressed as an 8-bit signed binary number using the sign-magnitude form as

$$\underset{\uparrow\ \text{Sign bit}}{0}\underbrace{0011001}_{\text{Magnitude bits}}$$

The decimal number −25 is expressed as

10011001

Notice that the only difference between +25 and −25 is the sign bit because the magnitude bits are in true binary for both positive and negative numbers.

In the sign-magnitude form, a negative number has the same magnitude bits as the corresponding positive number but the sign bit is a 1 rather than a zero.

Signed Numbers

1's Complement Form

Positive numbers in 1's complement form are represented the same way as the positive sign-magnitude numbers. Negative numbers, however, are the 1's complements of the corresponding positive numbers. For example, using eight bits, the decimal number −25 is expressed as the 1's complement of +25 (00011001) as

$$11100110$$

In the 1's complement form, a negative number is the 1's complement of the corresponding positive number.

Signed Numbers

2's Complement Form

Positive numbers in 2's complement form are represented the same way as in the sign-magnitude and 1's complement forms. Negative numbers are the 2's complements of the corresponding positive numbers. Again, using eight bits, let's take decimal number -25 and express it as the 2's complement of $+25$ (00011001). Inverting each bit and adding 1, you get

$$-25 = 11100111$$

In the 2's complement form, a negative number is the 2's complement of the corresponding positive number.

Signed Numbers

EXAMPLE 2–14

Express the decimal number −39 as an 8-bit number in the sign-magnitude, 1's complement, and 2's complement forms.

Solution

First, write the 8-bit number for +39.

00100111

In the *sign-magnitude form,* −39 is produced by changing the sign bit to a 1 and leaving the magnitude bits as they are. The number is

10100111

In the *1's complement form,* −39 is produced by taking the 1's complement of +39 (00100111).

11011000

In the *2's complement form,* −39 is produced by taking the 2's complement of +39 (00100111) as follows:

$$\begin{array}{rl} 11011000 & \text{1's complement} \\ +\quad\quad\;\; 1 & \\ \hline \mathbf{11011001} & \text{2's complement} \end{array}$$

Arithmetic Operations with Signed Numbers

1. Addition:

• The two numbers in an addition are the addend and the augend. The result is the sum. There are four cases that can occur when two signed binary numbers are added.

1. Both numbers positive.
2. Positive number with magnitude larger than negative number.
3. Negative number with magnitude larger than positive number.
4. Both numbers negative.
5. Equal and Opposite Numbers.

Arithmetic Operations with Signed Numbers

1. Addition:

Let's take one case at a time using 8-bit signed numbers as examples. The equivalent decimal numbers are shown for reference.

Both numbers positive:

(augend)	00000111	7	$(7)_{10} = (00000111)_2$
(addend)	+ 00000100	+ 4	$(4)_{10} = (00000100)_2$
(sum)	00001011	11	

The sum is positive and is therefore in true (uncomplemented) binary.

Positive number with magnitude larger than negative number:

	00001111	15	$(15)_{10} = (00001111)_2$
	+ 11111010	+ −6	$(6)_{10} = (00000110)_2$
Discard carry → 1	00001001	9	

The final carry bit is discarded. The sum is positive and therefore in true (uncomplemented) binary.

Arithmetic Operations with Signed Numbers

1. Addition:

Negative number with magnitude larger than positive number:

$$\begin{array}{r} 00010000 \\ +\ 11101000 \\ \hline 11111000 \end{array} \qquad \begin{array}{r} 16 \\ +\ -24 \\ \hline -8 \end{array} \qquad \begin{array}{l} (16)_{10} = (00010000)_2 \\ (24)_{10} = (00011000)_2 \end{array}$$

The sum is negative and therefore in 2's complement form.

Both numbers negative:

$$\begin{array}{rr} & 11111011 \\ & +\ 11110111 \\ \hline \text{Discard carry} \longrightarrow 1 & 11110010 \end{array} \qquad \begin{array}{r} -5 \\ +\ -9 \\ \hline -14 \end{array} \qquad \begin{array}{l} (5)_{10} = (00000101)_2 \\ (9)_{10} = (00001001)_2 \end{array}$$

The final carry bit is discarded. The sum is negative and therefore in 2's complement form.

Arithmetic Operations with Signed Numbers

2. Subtraction:

• Subtraction is a special case of addition. Subtraction is addition with the sign of the subtrahend changed.

• When you subtract two binary numbers with the 2’s complement method, it is important that both numbers have the same number of bits.

• The sign of a positive or negative binary number is changed by taking its 2’s complement.

• To subtract two signed numbers, take the 2’s complement of the subtrahend and add. Discard any final carry bit

Arithmetic Operations with Signed Numbers

2. Subtraction:

EXAMPLE 2–20

Perform each of the following subtractions of the signed numbers:

(a) 00001000 − 00000011 **(b)** 00001100 − 11110111

(c) 11100111 − 00010011 **(d)** 10001000 − 11100010

Solution

Like in other examples, the equivalent decimal subtractions are given for reference.

(a) In this case, $8 - 3 = 8 + (-3) = 5$.

	00001000	Minuend (+8)
	+ 11111101	2's complement of subtrahend (−3)
Discard carry →	**1 00000101**	Difference (+5)

(b) In this case, $12 - (-9) = 12 + 9 = 21$.

00001100	Minuend (+12)
+ 00001001	2's complement of subtrahend (+9)
00010101	Difference (+21)

Arithmetic Operations with Signed Numbers

2. Subtraction:

(c) In this case, $-25 - (+19) = -25 + (-19) = -44$.

	11100111	Minuend (−25)
	+ 11101101	2's complement of subtrahend (−19)
Discard carry	**1 11010100**	Difference (−44)

(d) In this case, $-120 - (-30) = -120 + 30 = -90$.

10001000	Minuend (−120)
+ 00011110	2's complement of subtrahend (+30)
10100110	Difference (−90)

Related Problem

Subtract 01000111 from 01011000.

Arithmetic Operations with Signed Numbers

3. Multiplication:

The numbers in a multiplication are the **multiplicand**, the **multiplier**, and the **product**. These are illustrated in the following decimal multiplication:

$$
\begin{array}{rl}
239 & \text{Multiplicand} \\
\times\ 123 & \text{Multiplier} \\
\hline
717 & \text{1st partial product } (3 \times 239) \\
478 & \text{2nd partial product } (2 \times 239) \\
+\ 239 & \text{3rd partial product } (1 \times 239) \\
\hline
29{,}397 & \text{Final product}
\end{array}
$$

Partial Products

Arithmetic Operations with Signed Numbers

3. Multiplication:

The sign of the product of a multiplication depends on the signs of the multiplicand and the multiplier according to the following two rules:

- **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 of binary multiplication are as follows:

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.

Arithmetic Operations with Signed Numbers

3. Multiplication:

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.

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.

Arithmetic Operations with Signed Numbers

3. Multiplication:

EXAMPLE 2–22

Multiply the signed binary numbers: 01010011 (multiplicand) and 11000101 (multiplier).

$(83)_{10} = (01010011)_2$ $(-59)_{10} = (11000101)_2$

Solution

Step 1: The sign bit of the multiplicand is 0 and the sign bit of the multiplier is 1. The sign bit of the product will be 1 (negative).

Step 2: Take the 2's complement of the multiplier to put it in true form.

$$11000101 \longrightarrow 00111011$$

Arithmetic Operations with Signed Numbers

3. Multiplication:

Step 3 and 4: The multiplication proceeds as follows. Notice that only the magnitude bits are used in these steps.

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

Arithmetic Operations with Signed Numbers

3. Multiplication:

Step 5: Since the sign of the product is a 1 as determined in step 1, take the 2's complement of the product.

1001100100001 ⟶ 0110011011111

Attach the sign bit

1 0110011011111

Related Problem

Verify the multiplication is correct by converting to decimal numbers and performing the multiplication.

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 and return the code to 8421. If a carry results when 6 is added, simply add the carry to the next 4-bit group.

BCD Addition

EXAMPLE 2–35

Add the following BCD numbers:

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

(c) 10000110 + 00010011 **(d)** 010001010000 + 010000010111

Solution

The decimal number additions are shown for comparison.

(a)

$$\begin{array}{r} 0011 \\ +\,0100 \\ \hline \mathbf{0111} \end{array} \qquad \begin{array}{r} 3 \\ +\,4 \\ \hline 7 \end{array}$$

(b)

$$\begin{array}{rr} 0010 & 0011 \\ +\,0001 & 0101 \\ \hline \mathbf{0011} & \mathbf{1000} \end{array} \qquad \begin{array}{r} 23 \\ +\,15 \\ \hline 38 \end{array}$$

(c)

$$\begin{array}{rr} 1000 & 0110 \\ +\,0001 & 0011 \\ \hline \mathbf{1001} & \mathbf{1001} \end{array} \qquad \begin{array}{r} 86 \\ +\,13 \\ \hline \mathbf{99} \end{array}$$

(d)

$$\begin{array}{rrr} 0100 & 0101 & 0000 \\ +\,0100 & 0001 & 0111 \\ \hline \mathbf{1000} & \mathbf{0110} & \mathbf{0111} \end{array} \qquad \begin{array}{r} 450 \\ +\,417 \\ \hline \mathbf{867} \end{array}$$

Note that in each case the sum in any 4-bit column does not exceed 9, and the results are valid BCD numbers.

BCD Addition

EXAMPLE 2–36

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)

$$
\begin{array}{rrll}
 & 1001 & & 9 \\
 & \underline{+\,0100} & & \underline{+4} \\
 & 1101 & \text{Invalid BCD number } (>9) & 13 \\
\underline{} & \underline{+\,0110} & \text{Add 6} & \\
\underbrace{\mathbf{0001}}_{1} & \underbrace{\mathbf{0011}}_{3} & \text{Valid BCD number} &
\end{array}
$$

(b)

$$
\begin{array}{rrll}
 & 1001 & & 9 \\
 & \underline{+\,1001} & & \underline{+\,9} \\
1 & 0010 & \text{Invalid because of carry} & 18 \\
\underline{} & \underline{+\,0110} & \text{Add 6} & \\
\underbrace{\mathbf{0001}}_{1} & \underbrace{\mathbf{1000}}_{8} & \text{Valid BCD number} &
\end{array}
$$

BCD Addition

(c)

$$
\begin{array}{rrll}
 & 0001 & 0110 & & 16 \\
+ & 0001 & \underline{0101} & & \underline{+\ 15} \\
 & 0010 & 1011 & \text{Right group is invalid (>9), left group is valid.} & 31 \\
 & & +\ 0110 & \text{Add 6 to invalid code. Add carry, 0001, to next group.} & \\
 & \underbrace{\mathbf{0011}}_{3} & \underbrace{\mathbf{0001}}_{1} & \text{Valid BCD number} &
\end{array}
$$

(d)

$$
\begin{array}{rrrll}
 & & 0110 & 0111 & & 67 \\
 & + & 0101 & \underline{0011} & & \underline{+\ 53} \\
 & & 1011 & 1010 & \text{Both groups are invalid (>9)} & 120 \\
 & + & 0110 & +\ 0110 & \text{Add 6 to both groups} & \\
 \underbrace{\mathbf{0001}}_{1} & & \underbrace{\mathbf{0010}}_{2} & \underbrace{\mathbf{0000}}_{0} & \text{Valid BCD number} &
\end{array}
$$

Related Problem

Add the BCD numbers: 01001000 + 00110100.

Hexadecimal Addition

For Hex numbers addition the following procedure is suggested

- Add the two hex digits in decimal.
- If the sum15 or less, it can be directly expressed as a hex digit.
- If the sum is greater than or equal to 16, subtract 16 and carry a 1 to the next digit position.

Example:-

$$\begin{array}{r} 5\ 8 \\ +\ 2\ 4 \\ \hline 7\ C \end{array} \qquad \begin{array}{r} 5\ 8 \\ +\ 4\ B \\ \hline A\ 3 \end{array} \qquad \begin{array}{r} 3\ A\ F \\ +\ 2\ 3\ C \\ \hline 5\ E\ B \end{array}$$

References

[1] Digital fundamentals / Thomas L. Floyd. —Eleventh edition.

[2]