Throughout these exercises, some conversions will be made between small binary, octal, decimal, and hexadecimal numbers. To avoid unnecessary clutter and distraction, these tables will be implicitly referenced.

| **Binary** | **Hexadecimal** | **Decimal** | **Binary** | **Hexadecimal** | **Decimal** |
|---|---|---|---|---|---|
| 0000 | 0x0 | 0 | 1000 | 0x8 | 8 |
| 0001 | 0x1 | 1 | 1001 | 0x9 | 9 |
| 0010 | 0x2 | 2 | 1010 | 0xA | 10 |
| 0011 | 0x3 | 3 | 1011 | 0xB | 11 |
| 0100 | 0x4 | 4 | 1100 | 0xC | 12 |
| 0101 | 0x5 | 5 | 1101 | 0xD | 13 |
| 0110 | 0x6 | 6 | 1110 | 0xE | 14 |
| 0111 | 0x7 | 7 | 1111 | 0xF | 15 |

| **Binary** | **Octal** | **Decimal** | **Binary** | **Octal** | **Decimal** |
|---|---|---|---|---|---|
| 000 | 00 | 0 | 100 | 04 | 4 |
| 001 | 01 | 1 | 101 | 05 | 5 |
| 010 | 02 | 2 | 110 | 06 | 6 |
| 011 | 03 | 3 | 111 | 07 | 7 |

EXERCISE 1.8 What is the largest unsigned 32-bit binary number?

SOLUTION

The largest 32-bit binary number would be a 32-digit number with 1 in every position.

$\Rightarrow$ 1111.1111.1111.1111.1111.1111.1111.1111

In decimal it's value is $2^{32} - 1$. That number is

$\Rightarrow 2^{32} - 1 = 4{,}294{,}967{,}295$

$\bowtie$

EXERCISE 1.10 What is the largest 32-bit binary number that can be represented with

(a) Unsigned numbers?

(b) Two's complement numbers?

SOLUTION

**(a)**

See EXERCISE 1.8.

In binary:

$\Rightarrow$ 1111.1111.1111.1111.1111.1111.1111.1111

In decimal:

$\Rightarrow 2^{32} - 1 = 4{,}294{,}967{,}295$

**(b)**

The range of a two's complement number is $-2^{N-1}, 2^{N-1} - 1$. The largest 32-bit binary number would have a 0 in the *msb* and the remaining digits would be 1. The decimal value is $2^{N-1} - 1$ or:

$\Rightarrow$ 0111.1111.1111.1111.1111.1111.1111.1111

$\Rightarrow 2^{31} - 1 = 2{,}197{,}483{,}647$

⋈

EXERCISE 1.12 What is the smallest (most negative) 32-bit binary number that can be represented with

(a) unsigned numbers?
(b) two's complement numbers?

SOLUTION

**(a)**

Axiomatically, the smallest unsigned number is zero. Every binary digit in the 32-digit binary number is zero.

$\Rightarrow$ 0000.0000.0000.0000.0000.0000.0000.0000

**(b)**

Two's complement binary numbers are signed. The smallest number that can be represented is the most negative. The *most significant digit* is the sign digit and it is 1 for negative numbers. The most negative two's complement number has zeros in every digit other than the sign digit. The decimal value is $-2^{N-1}$ or -2,147,483,648 for $N = 32$.

$\Rightarrow$ 1000.0000.0000.0000.0000.0000.0000.0000

⋈

EXERCISE 1.14 Convert the following unsigned binary numbers to decimal. Show your work.

(a) 1110
(b) 100100

(c) 1101.0111
(d) 011.1010.1010.0100

SOLUTION

In general, follow the method demonstrated in Figure 1.6 from the text. Break down digits into columns representing powers of 2.

**(a) 1110**

| **Power of 2** | **Column Name** | **Digit** | **Decimal Value** |
|---|---|---|---|
| $2^3$ | 8's column | 1 | $1x2^3$ |
| $2^2$ | 4's column | 1 | $1x2^2$ |
| $2^1$ | 2's column | 1 | $1x2^1$ |
| $2^0$ | 1's column | 0 | $0x2^0$ |

The decimal value is the result of adding the decimal value of each column that corresponds to the power of 2 it represents:

$$\begin{aligned} 1110 &= (1x2^3) + (1x2^2) + (1x2^1) + (0x2^0) \\ &= (1x8) + (1x4) + (1x2) + (0x1) \\ &= 8 + 4 + 2 + 0 \\ 1110 &= 14 \end{aligned}$$

$\Rightarrow$ 1110 = 14

**(b) 10.0100**

Follow the method demonstrated in Figure 1.6 from the text. Break down digits into columns representing powers of 2.

| **Power of 2** | **Column Name** | **Digit** | **Decimal Value** |
|---|---|---|---|
| $2^5$ | 32's column | 1 | $1x2^5$ |
| $2^4$ | 16's column | 0 | $0x2^4$ |
| $2^3$ | 8's column | 0 | $0x2^3$ |
| $2^2$ | 4's column | 1 | $1x2^2$ |
| $2^1$ | 2's column | 0 | $0x2^1$ |
| $2^0$ | 1's column | 0 | $0x2^0$ |

The decimal value is the result of adding the decimal value of each column that corresponds to the power of 2 it represents:

$$\begin{aligned} 10.0100 &= (1x2^5) + (0x2^4) + (0x2^3) + (1x2^2) + (0x2^1) + (0x2^0) \\ &= (1x32) + (0x16) + (0x8) + (1x4) + (0x2) + (0x1) \\ &= 32 + 0 + 0 + 4 + 0 + 0 \\ 10.0100 &= 36 \end{aligned}$$

$\Rightarrow$ 10.0100 = 36

**(c) 1101.0111**

Follow the method demonstrated in Figure 1.6 from the text. Break down digits into columns representing powers of 2.

| **Power of 2** | **Column Name** | **Digit** | **Decimal Value** |
|---|---|---|---|
| $2^7$ | 128's column | 1 | $1x2^7$ |
| $2^6$ | 64's column | 1 | $1x2^6$ |
| $2^5$ | 32's column | 0 | $0x2^5$ |
| $2^4$ | 16's column | 1 | $1x2^4$ |
| $2^3$ | 8's column | 1 | $0x2^3$ |
| $2^2$ | 4's column | 1 | $1x2^2$ |
| $2^1$ | 2's column | 1 | $1x2^1$ |
| $2^0$ | 1's column | 1 | $1x2^0$ |

The decimal value is the result of adding the decimal value of each column that corresponds to the power of 2 it represents:

$$\begin{aligned}
1101.0111 &= (1x2^7) + (1x2^6) + (0x2^5) + (1x2^4) + (0x2^3) + (1x2^2) + (1x2^1) + (1x2^0) \\
&= (1x128) + (1x64) + (0x32) + (1x16) + (0x8) + (1x4) + (1x2) + (1x1) \\
&= 128 + 64 + 0 + 16 + 0 + 4 + 2 + 1 \\
1101.0111 &= 215
\end{aligned}$$

$\Rightarrow 1101.0111 = 215$

**(d) 011.1010.1010.0100**

Follow the method demonstrated in Figure 1.6 from the text. Break down digits into columns representing powers of 2.

| **Power of 2** | **Column Name** | **Digit** | **Decimal Value** |
|---|---|---|---|
| $2^{14}$ | 16384's column | 0 | $0x2^{14}$ |
| $2^{13}$ | 8192's column | 1 | $1x2^{13}$ |
| $2^{12}$ | 4096's column | 1 | $1x2^{12}$ |
| $2^{11}$ | 2048's column | 1 | $1x2^{11}$ |
| $2^{10}$ | 1024's column | 0 | $0x2^{10}$ |
| $2^9$ | 512's column | 1 | $1x2^9$ |
| $2^8$ | 256's column | 0 | $0x2^8$ |
| $2^7$ | 128's column | 1 | $1x2^7$ |
| $2^6$ | 64's column | 0 | $0x2^6$ |
| $2^5$ | 32's column | 1 | $1x2^5$ |
| $2^4$ | 16's column | 0 | $0x2^4$ |
| $2^3$ | 8's column | 0 | $0x2^3$ |
| $2^2$ | 4's column | 1 | $1x2^2$ |
| $2^1$ | 2's column | 0 | $0x2^1$ |

$2^0$ 1's column 0 $0x2^0$

The decimal value is the result of adding the decimal value of each column that corresponds to the power of 2 it represents:

$$\begin{aligned}
011.1010.1010.0100 &= (0x2^{14}) + (1x2^{13}) + (1x2^{12}) + (1x2^{11}) + (0x2^{10}) + (1x2^{9}) + (0x2^{8}) \\
&\quad + (1x2^{7}) + (0x2^{6}) + (1x2^{5}) + (0x2^{4}) + (0x2^{3}) + (1x2^{2}) + (0x2^{1}) \\
&\quad + (0x2^{0}) \\
&= (0x16384) + (1x8192) + (1x4096) + (1x2048) + (0x1024) + (1x512) \\
&\quad + (0x256) \\
&= 0 + 8192 + 4096 + 2048 + 0 + 512 + 0 + 128 + 0 + 32 + 0 + 0 + 4+ \\
&\quad 0 + 0 \\
011.1010.1010.0100 &= 15,012
\end{aligned}$$

⇒ 011.1010.1010.0100 = 15,012

⋈

EXERCISE 1.16 Repeat EXERCISE 1.14 but convert to hexadecimal.

(a) 1110
(b) 10.0100
(c) 1101.0111
(d) 011.1010.1010.0100

SOLUTION

Each 4-bit nibble corresponds to a hexadecimal digit. Convert each binary number to a multiple of 4 bits by prefixing leading zeros in the *most significant* positions. Then convert each 4-bit nibble to a hexadecimal digit and assemble in the correct order.

**(a) 1110**

By inspection,

⇒ 1110 = 0xE

**(b) 10.0100**

Using sign extension convert the 6-bit binary number to an 8-bit binary number:

$$10.0100 = 0010.0100$$

By inspection, each 4-bit nibble is equal to a single hexadecimal digit:

$$\begin{aligned}
0010 &= 0\text{x}2 \\
0100 &= 0\text{x}4
\end{aligned}$$

Now assemble the hexadecimal digits that correspond to each nibble:

$\Rightarrow$ 10.0100 = 0x24

**(c) 1101.0111**

$$1101 = \text{0xD}$$
$$0111 = \text{0x7}$$

Now assemble the hexadecimal digits that correspond to each nibble:

$\Rightarrow$ 1101.0111 = 0xD7

**(d) 011.1010.1010.0100**

Using sign extension, extend this binary number to 16 bits so it is an even multiple of 4 binary digits:

$$011.1010.1010.0100 = 0011.1010.1010.0100$$

By inspection, each 4-bit nibble is equal to a single hexadecimal digit:

$$0011 = \text{0x3}$$
$$1010 = \text{0xA}$$
$$1010 = \text{0xA}$$
$$0100 = \text{0x4}$$

Now assemble the hexadecimal digits that correspond to each nibble:

$\Rightarrow$ 011.1010.1010.0100 = 0x3AA4

$\bowtie$

EXERCISE 1.18 Convert the following hexadecimal numbers to decimal. Show your work.
(a) 0x4E
(b) 0x7C
(c) 0x4D3A
(d) 0x403FB001

SOLUTION

In general, the conversion approach requires taking the sum of each of the decimal values of each hexadecimal digit. The decimal values are represented by $16^{N-1}$ where $N$ is the digit number (column).

**(a) 0x4E**

| Power of 16 | Column Name | Digit | Decimal Value |
|---|---|---|---|

| | | | |
|---|---|---|---|
| $16^1$ | 16's column | 0x4 | 4 |
| $16^0$ | 1's column | 0xE | 14 |

$$\begin{aligned} \text{0x4E} &= (4x16^1) + (14x16^0) \\ &= (4x16) + (14x1) \\ \text{0x4E} &= 78 \end{aligned}$$

⇒ 0x4E = 78

**(b) 0x7C**

| Power of 16 | Column Name | Digit | Decimal Value |
|---|---|---|---|
| $16^1$ | 16's column | 0x7 | 7 |
| $16^0$ | 1's column | 0xC | 12 |

$$\begin{aligned} \text{0x7C} &= (7x16^1) + (12x16^0) \\ &= (7x16) + (12x1) \\ \text{0x7C} &= 124 \end{aligned}$$

⇒ 0x7C = 124

**(c) 0xED3A**

| Power of 16 | Column Name | Digit | Decimal Value |
|---|---|---|---|
| $16^3$ | 4096's column | 0xE | 14 |
| $16^2$ | 256's column | 0xD | 13 |
| $16^1$ | 16's column | 0x3 | 3 |
| $16^0$ | 1's column | 0xA | 10 |

$$\begin{aligned} \text{0xED3A} &= (14x16^3) + (13x16^2) + (3x16^1) + (10x16^0) \\ &= (14x4096) + (13x256) + (3x16) + (10x1) \\ &= 57,344 + 3,328 + 48 + 10 \\ \text{0xED3A} &= 60,730 \end{aligned}$$

⇒ 0xED3A = 60,730

**(d) 0x403FB001**

| Power of 16 | Column Name | Digit | Decimal Value |
|---|---|---|---|
| $16^7$ | 268,434,456's column | 0x4 | 4 |
| $16^6$ | 16,777,216's column | 0x0 | 0 |

| | | | |
|---|---|---|---|
| $16^5$ | 1,048,576's column | 0x3 | 3 |
| $16^4$ | 65536's column | 0xF | 15 |
| $16^3$ | 4096's column | 0xB | 11 |
| $16^2$ | 256's column | 0x0 | 0 |
| $16^1$ | 16's column | 0x0 | 0 |
| $16^0$ | 1's column | 0x1 | 1 |

$$\begin{aligned}
\text{0x403FB001} &= (4x16^7) + (0x16^6) + (3x16^5) + (15x16^4) + (11x16^3) + (0x16^2) + (0x16^1) \\
&\quad + (1x16^0) \\
&= (4x268,435,456) + (0x16,777,216) + (3x1,048,576)(15x65,536) + (11x4096) \\
&\quad + (0x256) + (0x16) + (1x1) \\
&= 1,073,741,824 + 0 + 3,145,728 + 983,040 + 45,056 + 0 + 0 + 1 \\
\text{0x403FB001} &= 1,077,915,649
\end{aligned}$$

$\Rightarrow$ 0x403FB001 = 1,077,915,649

$\bowtie$

EXERCISE 1.20 Repeat EXERCISE 1.18 but convert to unsigned binary.

(a) 0x4E
(b) 0x7C
(c) 0xED3A
(d) 0x403FB001

SOLUTION

In general, convert each hexadecimal digit to it's 4-bit binary representation and assemble in *most significant* order.

**(a) 0x4E**

By inspection, 0x4E breaks down to the 4-bit nibbles

$$\begin{aligned}
\text{0x4} &= 0100 \\
\text{0xE} &= 1110
\end{aligned}$$

So the unsigned binary representation is:

$\Rightarrow$ 0x4E = 0100.1110

**(b) 0x7C**

By inspection, 0x7C breaks down to the 4-bit nibbles

$$\begin{aligned}
\text{0x7} &= 0111 \\
\text{0xC} &= 1100
\end{aligned}$$

So the unsigned binary representation is:

$\Rightarrow$ 0x7C = 0111.1100

**(c) 0xED3A**

By inspection, 0xED3A breaks down to the 4-bit nibbles

$$\begin{aligned} \text{0xE} &= 1110 \\ \text{0xD} &= 1101 \\ \text{0x3} &= 0011 \\ \text{0xA} &= 1010 \end{aligned}$$

So the unsigned binary is

$\Rightarrow$ 0xED3A = 1110.1101.0011.1010

**(d) 0x403FB001**

By inspection, 0x403FB001 breaks down to the 4-bit nibbles

$$\begin{aligned} \text{0x4} &= 0100 \\ \text{0x0} &= 0000 \\ \text{0x3} &= 0011 \\ \text{0xF} &= 1111 \\ \text{0xB} &= 1011 \\ \text{0x0} &= 0000 \\ \text{0x0} &= 0000 \\ \text{0x1} &= 0001 \end{aligned}$$

So the unsigned binary is

$\Rightarrow$ 0x403FB001 = 0100.0000.0011.1111.1011.0000.0000.0001

$\bowtie$

EXERCISE 1.22 Convert the following two's complement binary numbers to decimal.
(a) 1110
(b) 100011
(c) 01001110
(d) 10110101

SOLUTION

**(a) 1110**

1110 is negative. Compute it's decimal value by summing the decimal value of each column as demonstrated in Figure 1.6 of the text.

| **Power of 2** | **Column Name** | **Digit** | **Decimal Value** |
|---|---|---|---|
| $2^3$ | 8's column | 1 | $1x2^3$ |
| $2^2$ | 4's column | 1 | $1x2^2$ |
| $2^1$ | 2's column | 1 | $1x2^1$ |
| $2^0$ | 1's column | 0 | $0x2^0$ |

$$\begin{aligned}
1110 &= (-1x2^3) + (1x2^2) + (1x2^1) + (0x2^0) \\
&= (-1x8) + (1x4) + (1x2) + (0x1) \\
&= -8 + 4 + 2 + 0 \\
1110 &= -2
\end{aligned}$$

⇒ 1110 signed = -2

Alternatively, find it's magnitude by converting to unsigned binary. Take the two's complement by inverting the bits and adding 1.

Invert the bits: 1110 ⇒ 0001
Add 1: $0001 + 0001 = 0010$

Again, summing the columns in the table above...

$$\begin{aligned}
0010 &= (0x2^3) + (0x2^2) + (1x2^1) + (0x2^0) \\
&= (0x8) + (0x4) + (1x2) + (0x1) \\
&= 0 + 0 + 2 + 0 \\
0010 &= 2
\end{aligned}$$

But since the signed binary number 1110 is negative, we apply the sign to the magnitude 2.

⇒ 1110 signed = -2

**(b) 100011**

100011 is negative. Compute it's decimal value by summing the decimal value of each column.

| **Power of 2** | **Column Name** | **Digit** | **Decimal Value** |
|---|---|---|---|
| $2^5$ | 32's column | 1 | $1x2^5$ |
| $2^4$ | 16's column | 0 | $0x2^4$ |
| $2^3$ | 8's column | 0 | $0x2^3$ |
| $2^2$ | 4's column | 0 | $0x2^2$ |
| $2^1$ | 2's column | 1 | $1x2^1$ |
| $2^0$ | 1's column | 1 | $1x2^0$ |

$$\begin{aligned}
100011 &= (-1x2^5) + (0x2^4) + (0x2^3) + (0x2^2) + (1 + 2^1) + (1 + 2^0)\\
&= (-1x32) + (0x16) + (0x8) + (0x4) + (1 + 2) + (1 + 1)\\
&= -32 + 0 + 0 + 0 + 2 + 1\\
100011 &= -29
\end{aligned}$$

⇒ 100011 = -29

Alternatively, find it's magnitude by taking the two's complement and apply the sign to the decimal result.
Take the two's complement by inverting and adding 1:

$$\begin{aligned}
\neg\, 10.0011 &= 01.1100\\
01.1100 + 00.0001 &= 01.1101
\end{aligned}$$

Again, summing the columns in the table above...

$$\begin{aligned}
01.1101 &= (0x2^5) + (1x2^4) + (1x2^3) + (1x2^2) + (0x2^1) + (1x2^0)\\
&= (0x32) + (1x16) + (1x8) + (1x4) + (0x2) + (1x1)\\
&= 0 + 16 + 8 + 4 + 0 + 1\\
01.1101 &= 29
\end{aligned}$$

Apply the sign ⇒ $-29$.

⇒ 10.0011 = -29

**(c) 0100.1110**

0100.1110 is positive. Positive two's complement binary is equivalent to unsigned binary. Convert to decimal by summing the decimal values of each column:

| **Power of 2** | **Column Name** | **Digit** | **Decimal Value** |
|---|---|---|---|
| $2^7$ | 128's column | 0 | $0x2^5$ |
| $2^6$ | 64's column | 1 | $1x2^5$ |
| $2^5$ | 32's column | 0 | $0x2^5$ |
| $2^4$ | 16's column | 0 | $0x2^4$ |
| $2^3$ | 8's column | 1 | $1x2^3$ |
| $2^2$ | 4's column | 1 | $1x2^2$ |
| $2^1$ | 2's column | 1 | $1x2^1$ |
| $2^0$ | 1's column | 0 | $0x2^0$ |

$$\begin{aligned}
0100.1110 &= (0x2^7) + (1x2^6) + (0x2^5) + (0x2^4) + (1x2^3) + (1x2^2) + (1x2^1) + (0x2^0)\\
&= (0x128) + (1x64) + (0x32) + (0x16) + (1x8) + (1x4) + (1x2) + (0x1)\\
&= 0 + 64 + 0 + 0 + 8 + 4 + 2 + 0\\
0100.1110 &= 78
\end{aligned}$$

$\Rightarrow$ 0100.1110 = 78

**(d) 1011.0101**

1011.0101 is negative. Compute it's decimal value by summing the decimal value of each column.

| Power of 2 | Column Name | Digit | Decimal Value |
|---|---|---|---|
| $2^7$ | 128's column | 1 | $1x2^7$ |
| $2^6$ | 64's column | 0 | $0x2^6$ |
| $2^5$ | 32's column | 1 | $1x2^5$ |
| $2^4$ | 16's column | 1 | $1x2^4$ |
| $2^3$ | 8's column | 0 | $0x2^3$ |
| $2^2$ | 4's column | 1 | $1x2^2$ |
| $2^1$ | 2's column | 0 | $0x2^1$ |
| $2^0$ | 1's column | 1 | $1x2^0$ |

$$\begin{aligned}
1011.0101 &= (-1x2^7) + (0x2^6) + (1x2^5) + (1 + 2^4) + (0x2^3) + (1x2^2) + (0x2^1) + (1x2^0) \\
&= (-1x128) + (0x64) + (1x32) + (1 + 16) + (0x8) + (1x4) + (0x2) + (1x1) \\
&= -128 + 0 + 32 + 16 + 0 + 4 + 0 + 1 \\
1011.0101 &= -75
\end{aligned}$$

$\Rightarrow$ 1011.0101 = −75

Alternatively, find it's magnitude by taking the two's complement and applying the sign. Take the twos complement by inverting and adding 1:

$$\begin{aligned}
\neg\, 1011.0101 &= 0100.1010 \\
0100.1010 + 0000.0001 &= 0100.1011
\end{aligned}$$

Again, summing the columns in the table above. . .

$$\begin{aligned}
0100.1011 &= (0x2^7) + (1x2^6) + (0x2^5) + (0x2^4) + (1x2^3) + (0x2^2) + (1x2^1) + (1x2^0) \\
&= (0x128) + (1x64) + (0x32) + (0x16) + (1x8) + (0x4) + (1x2) + (1x1) \\
&= 0 + 64 + 0 + 0 + 8 + 0 + 2 + 1 \\
0100.1011 &= 75
\end{aligned}$$

Now apply the sign.

$\Rightarrow$ 1011.0101 = −75

⋈

EXERCISE 1.26 Convert the following decimal numbers to unsigned binary numbers.

(a) 14
(b) 52
(c) 339
(d) 711

SOLUTION

In general we can choose from one of the two methods demonstrated in Example 1.5 from the text.

**(a) 14**

Divide by the highest power of 2. The quotient will be the binary digit. Repeat the process with the remainder for each power of 2.

$$\begin{aligned} 14/8 &= \underline{1}r6 \\ 6/4 &= \underline{1}r2 \\ 2/2 &= \underline{1}r0 \\ 0/1 &= \underline{0}r0 \end{aligned}$$

The binary number is derived by using the quotient in the column for each power of 2. The result corresponds to the value of columns 8, 4, 2, 1.

$\Rightarrow 14 = 1110$

**(b) 52**

Solve by repeating the process used in **(a)**. The highest power of 2 that is less than or equal to 52 is $2^5 = 32$.

$$\begin{aligned} 52/32 &= \underline{1}r20 \\ 20/16 &= \underline{1}r4 \\ 4/8 &= \underline{0}r4 \\ 4/4 &= \underline{1}r0 \\ 0/2 &= \underline{0}r0 \\ 0/1 &= \underline{0}r0 \end{aligned}$$

Now assemble each digit from the quotients in the columns corresponding to the respective powers of 2 in *most significant* order.

$\Rightarrow 52 = 11.0100$

**(c) 339**

Solve by repeating the process used in **(a)**. The highest power of 2 that is less than or equal to 339 is $2^8 = 256$.

$$
\begin{aligned}
339/256 &= \underline{1}r83 \\
83/128 &= \underline{0}r83 \\
83/64 &= \underline{1}r19 \\
19/32 &= \underline{0}r19 \\
19/16 &= \underline{1}r3 \\
3/8 &= \underline{0}r3 \\
3/4 &= \underline{0}r3 \\
3/2 &= \underline{0}r1 \\
1/1 &= \underline{1}r0
\end{aligned}
$$

Now assemble each digit from the quotients in the columns corresponding to the respective powers of 2 in *most significant* order.

$\Rightarrow$ 339 = 1.0101.0011

**(d) 711**

Solve by repeating the process used in previous steps. The highest power of 2 that is less than or equal to 711 is $2^9 = 512$.

$$
\begin{aligned}
711/512 &= \underline{1}r199 \\
199/256 &= \underline{0}r199 \\
199/128 &= \underline{1}r71 \\
71/64 &= \underline{1}r7 \\
7/32 &= \underline{0}r7 \\
7/16 &= \underline{0}r7 \\
7/8 &= \underline{0}r7 \\
7/4 &= \underline{1}r3 \\
3/2 &= \underline{1}r1 \\
1/1 &= \underline{1}r0
\end{aligned}
$$

Now assemble each digit from the quotients in the columns corresponding to the respective powers of 2 in *most significant* order.

$\Rightarrow$ 711 = 10.1100.0111

$\bowtie$

EXERCISE 1.28 Repeat EXERCISE 1.26 but convert to hexadecimal.

(a) 14
(b) 52

(c) 339
(d) 711

SOLUTION

In general choose from one of the methods demonstrated in Example 1.5 from the text.

**(a) 14**

The highest power of 16 less than or equal to 14 is $16^0 = 1$.

$$14/1 = \underline{14}r0$$

By inspection, the result is

$\Rightarrow 14 = $ 0xE

**(b) 52**

The highest power of 16 less than or equal to 52 is $16^1 = 16$.

$$\begin{aligned} 52/16 &= \underline{3}r4 \\ 4/1 &= \underline{4}r0 \end{aligned}$$

Combine the hexadecimal digits on *most significant* order:

$\Rightarrow 52 = $ 0x34

**(c) 339**

The highest power of 16 less than or equal to 339 is $16^2 = 256$.

$$\begin{aligned} 339/256 &= \underline{1}r83 \\ 83/16 &= \underline{5}r3 \\ 3/1 &= \underline{3}r0 \end{aligned}$$

Combine the hexadecimal digits in *most significant* order.

$\Rightarrow 339 = $ 0x153

**(d) 711**

The highest power of 16 less than or equal to 711 is $16^2 = 256$.

$$\begin{aligned} 711/256 &= \underline{2}r199 \\ 199/16 &= \underline{12}r7, \text{0xC by inspection.} \\ 7/1 &= \underline{7}r0 \end{aligned}$$

Combine the hexadecimal digits in *most significant* order.

$\Rightarrow 711 = $ 0x2C7

⋈

EXERCISE 1.30 Convert the following decimal numbers to 8-bit two's complement numbers or indicate that the decimal number would overflow the range.

(a) 24
(b) -59
(c) 128
(d) -150
(e) 127

SOLUTION

In general, choose from one of the methods demonstrated in Example 1.5 from the text to convert the decimal number to unsigned binary. Then take the two's complement by inverting each binary digit and adding 1.

The range of a signed binary number is, in general, $(-2^{N-1}, 2^{N-1}-1)$. In this case with 8-bit binary numbers as a given, the range is (-128,127).

**(a) 24**

The highest power of 2 less than or equal to 24 is $2^4 = 16$.

$$\begin{aligned} 24/16 &= \underline{1}r8 \\ 8/8 &= \underline{1}r0 \\ 0/4 &= \underline{0}r0 \\ 0/2 &= \underline{0}r0 \\ 0/1 &= \underline{0}r0 \end{aligned}$$

Assemble the digits to yield the unsigned binary result:

⇒ 1.1000

Now perform sign extension to get the 8-bit result. Since 24 is a positive number, the sign digit is 0. The result is:

$\boxed{\Rightarrow 24 = 0001.1000}$

**(b) -59**

Repeatedly divide the magnitude by the highest divisible power of 2 to convert to unsigned binary. The highest power of 2 less than or equal to 59 is $2^5 = 32$.

$$\begin{aligned} 59/32 &= \underline{1}r27 \\ 27/16 &= \underline{1}r11 \\ 11/8 &= \underline{1}r3 \\ 3/4 &= \underline{0}r3 \\ 3/2 &= \underline{1}r1 \\ 1/1 &= \underline{1}r0 \end{aligned}$$

Now assemble the digits in order to get the unsigned binary result $\Rightarrow$ 11.1011.

Next we can proceed in one of two ways:

1. Unsigned sign extension, then take two's complement.
2. Take two's complement, then sign extension.

**(1)** Perform sign extension of unsigned result $\Rightarrow$ 0011.1011.

Take the two's complement by inverting and adding 1:

$$\neg\, 0011.1011 = 1100.0100$$
$$1100.0100 + 0000.0001 = 1100.0101$$

This yields the result:

$\Rightarrow$ -59 = 1100.0101

**(2)** Take the two's complement by inverting and adding 1:

$$\neg\, 11.1011 = 00.0100$$
$$00.0100 + 00.0001 = 00.0101$$

Now extending to 8 digits using 1 as the sign bit yields:

$\Rightarrow$ -59 = 1100.0101

**(c) 128**

The range as stated earlier is (-128,127). This number is outside the range of an 8-bit two's complement signed binary number. It overflows.

**(d) -150**

The range as stated earlier is (-128,127). This number is outside the range of an 8-bit two's complement signed binary number. It overflows.

**(e) 127**

This number is inside the range of an 8-bit two's complement signed binary number. It does not overflow. The number is positive so the unsigned result of converting to binary will be correct.

Divide repeatedly by the highest power of 2 less than or equal to 127. The highest power of 2 is $2^6 = 64$.

$$127/64 = \underline{1}r63$$
$$63/32 = \underline{1}r31$$
$$31/16 = \underline{1}r15$$
$$15/8 = \underline{1}r7$$
$$7/4 = \underline{1}r3$$
$$3/2 = \underline{1}r1$$
$$1/1 = \underline{1}r0$$

Assemble the digits in column order to get the unsigned binary result $\Rightarrow$ 111.1111. Now perform sign extension to get the 8-bit binary number. The sign bit is zero:

$\Rightarrow$ 127 = 0111.1111

⋈

EXERCISE 1.34 Convert the following 4-bit two's complement numbers to 8-bit two's complement numbers.
(a) 0111
(b) 1001

Use sign extension—copy the sign bit to the *most significant bit* positions.

**(a) 0111**

The sign bit is 0. Therefore, copy the 0 to the four *most significant bit* positions to yield:

$\Rightarrow$ 0111 = 0000.0111

**(b) 1001**

The sign bit is 1. Therefore copy 1 to the four *most significant bit* positions to yield:

$\Rightarrow$ 1001 = 1111.1001

⋈

EXERCISE 1.36 Repeat EXERCISE 1.34 if the numbers are unsigned rather than two's complement.
(a) 0111
(b) 1001

SOLUTION

In this case, we simply copy 0 into the 4 *most significant bit* positions.

By inspection,

$\Rightarrow$ 0111 = 0000.0111
$\Rightarrow$ 1001 = 0000.1001

⋈

EXERCISE 1.38 Base 8 is referred to as octal. Convert each of the numbers from EXERCISE 1.26 to octal.
(a) 14
(b) 52

(c) 339
(d) 711

SOLUTION

In general, repeatedly divide the number to be converted by the largest power of 8 that is less than or equal to the number. Each division will yield a whole number quotient and a remainder. Each successive division will operate on the remainder of the previous division. This requires the next lower power of 8 to be used. The quotients will then be assembled in *most significant* order to form the converted number. This is the first method demonstrated in Example 1.5 of the text.

**(a) 14**

The highest power of 8 less than or equal to 14 is $8^1 = 8$.

$$\begin{aligned} 14/8 &= \underline{1}r6 \\ 6/1 &= \underline{6}r0 \end{aligned}$$

Now assemble the digits and epress the answer in octal format.

$\Rightarrow 14 = 016$

**(b) 52**

The highest power of 8 less than or equal to 52 is $8^1 = 8$.

$$\begin{aligned} 52/8 &= \underline{6}r4 \\ 4/1 &= \underline{4}r0 \end{aligned}$$

Now assemble the digits in octal format.

$\Rightarrow 52 = 064$

**(c) 339**

The highest power of 8 less than or equal to 339 is $8^2 = 64$.

$$\begin{aligned} 339/64 &= \underline{5}r19 \\ 19/8 &= \underline{2}r3 \\ 3/1 &= \underline{3}r0 \end{aligned}$$

Now assemble the digits in octal format.

$\Rightarrow 339 = 0523$

**(d) 711**

The highest power of 8 lelss than or equal to 711 is $8^3 = 512$.

$$\begin{aligned} 711/512 &= \underline{1}r199 \\ 199/64 &= \underline{3}r7 \\ 7/8 &= \underline{0}r7 \\ 7/1 &= \underline{7}r0 \end{aligned}$$

Now assemble the digits in octal format.

$\Rightarrow$ 711 = 01307

⋈

Exercise 1.40 Convert each of the following octal numbers to binary, hexadecimal, and decimal.
(a) 023
(b) 045
(c) 0371
(d) 02560

solution

Each octal digit represents a 3-digit binary number. By inspection, substitute the binary number for the octal digit to convert to binary.

The binary result from the previous step should be extended to an 8-bin binary number. Each 4-bit nibble represents a single hexadecimal digit. Combine the digits to find the result.

Convert the octal to decimal by multiplying the octal digit by the power of 8 for that column.

**a) 023**

Convert to binary: By inspection, 02 = 010 and 03 = 011. The binary result is the combination of the binary numbers in the proper order.

$\Rightarrow$ 023 = 010.011

Convert to hexadecimal: Extend the previous result from 6 binary digits to 8 binary digits by prefixing zeros in the *most significant bit* positions. This yields 0001.0011.

Each 4-bit nibble represents a single hexadecimal digit. By inspection, we can write:

$$\begin{aligned} 0001 &= \text{0x1} \\ 0011 &= \text{0x3} \end{aligned}$$

The hexadecimal result is the combination of the two hexadecimal digits in *most significant digit* order.

$\Rightarrow$ 023 = 0x13

To convert the octal value to decimal, use the method demonstrated in Figure 1.6 from the text:

$$\begin{aligned} 023 &= (2x8^1) + (3x8^0) \\ &= (2x8) + (3x1) \\ &= 16 + 3 \\ 023 &= 19 \end{aligned}$$

$\Rightarrow$ 023 = 19

**b) 045**

Convert to binary: By inspection, 04 = 100 and 05 = 101. The binary result is the combination of the binary numbers in the proper order.

$\Rightarrow$ 045 = 100.101

Convert to hexadecimal: Extend the previous result from 6 binary digits to 8 binary digits by prefixing zeros in the *most significant bit* positions. This yields 0010.0101.

Each 4-bit nibble represents a single hexadecimal digit. By inspection, we can write:

$$\begin{aligned} 0010 &= \text{0x2} \\ 0101 &= \text{0x5} \end{aligned}$$

The hexadecimal result is the combination of the two hexadecimal digits in *most significant digit* order.

$\Rightarrow$ 045 = 0x25

To convert the octal value to decimal, use the method demonstrated in Figure 1.6 from the text:

$$\begin{aligned} 045 &= (4x8^1) + (5x8^0) \\ &= (4x8) + (5x1) \\ &= 32 + 5 \\ 045 &= 37 \end{aligned}$$

$\Rightarrow$ 045 = 37

**c) 0371**

Convert to binary: By inspection, 03 = 011, 07 = 111, and 01 = 001. The binary result is the combination of the binary numbers in *most significant* order.

$\Rightarrow$ 0371 = 011.111.001

Convert to hexadecimal: The previous result was represented as 9 binary digits with a leading 0 in the *most significant bit* position just for clarity. Removing that and reformatting gives us the binary number 1111.1001. Each 4-bit nibble represents a single hexadecimal digit. By inspection, we can write:

$$\begin{aligned} 1111 &= 0\text{xF} \\ 1001 &= 0\text{x9} \end{aligned}$$

The hexadecimal result is the combination of the two hexadecimal digits in *most significant digit* order.

$\Rightarrow$ 0xF9

To convert the octal value to decimal, use the method demonstrated in Figure 1.6 of the text:

$$\begin{aligned} 0371 &= (3x8^2) + (7x8^1) + (1x8^0) \\ &= (3x64) + (7x8) + (1x1) \\ &= 192 + 56 + 1 \\ 0371 &= 249 \end{aligned}$$

$\Rightarrow$ 0371 = 249

**d) 02560**

Convert to binary: By inspection:

$$\begin{aligned} 02 &= 010 \\ 05 &= 101 \\ 06 &= 110 \\ 00 &= 000 \end{aligned}$$

The binary result is the combination of the binary numbers in *most significant* order.

$\Rightarrow$ 010.101.110.000

Convert to hexadecimal: The previous result is represented in groupings of three binary digits only to illustrate the relationship to the original octal digits. Represented in groups of 4-bit nibbles to represent hexadecimal digits yields 0101.0111.0000. By inspection, we can write:

$$\begin{aligned} 0101 &= 0\text{x5} \\ 0111 &= 0\text{x7} \\ 0000 &= 0\text{x0} \end{aligned}$$

The hexadecimal result is the combination of the three hexadecimal digits in *most significant digit* order.

⇒ 0x570

To convert the octal value to decimal, use the method demonstrated in Figure 1.6 of the text:

$$\begin{aligned} 02560 &= (2x8^3) + (5x8^2) + (6x8^1) + (0x8^0) \\ &= (2x512) + (5x64) + (6x8) + (0x1) \\ &= 1024 + 320 + 48 + 0 \\ 02560 &= 1392 \end{aligned}$$

⇒ $02560 = 1392$

⋈

EXERCISE 1.42 How many 7-bit two's complement numbers are greater than 0? How many are less than 0?

SOLUTION

The range of a two's complement number giving $N$ binary digits can be stated generally as $-2^{N-1}, 2^{N-1} - 1$. Given that $N = 7$, the largest positive number is $2^{N-1} - 1 = 63$. The largest negative number is $2^{N-1} = 64$.

⋈

EXERCISE 1.44 How many bytes are in a 64-bit word?

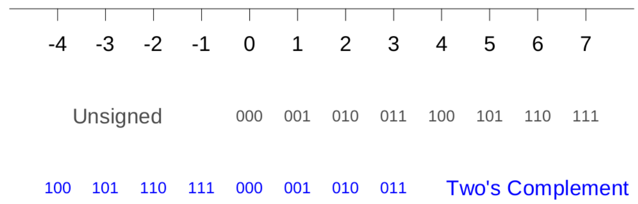
SOLUTION

Axiomatically, a byte is an 8-bit group of binary digits. A 64-bit word therefore contains $64/8 = 8$ bytes.

⋈

EXERCISE 1.50 Draw a number line analogous to Figure 1.11 for 3-bit unsigned and two's complement numbers.

SOLUTION




⋈

EXERCISE 1.52 Perform the following additions of unsigned binary numbers. Indicate whether or not the sum overflows a 4-bit result.

(a) 1001 + 0100
(b) 1101 + 1011

SOLUTION

**(a) 1001 + 0100**

$1001 + 0100 = 1101$. The sum does not overflow a 4-bit result.

**(b) 1101 + 1011**

$1101 + 1011 = 11000$. The sum does overflow a 4-bit result.

⋈

EXERCISE 1.54 Repeat EXERCISE 1.52, assuming that the binary numbers are in two's complement form.

(a) 1001 + 0100
(b) 1101 + 1011

SOLUTION

**(a) 1001 + 0100**

$1001 + 0100 = 1101$. The sum does not overflow a 4-bit two's complement signed binary number.

**(b) 1101 + 1011**

$1101 + 1011 = 1000$. The sum does not overflow a 4-bit two's complement signed binary number.

⋈

EXERCISE 1.56 Convert the following decimal numbers to 6-bit two's complement binary numbers and add them. Indicate whether or not the sum overflows a 6-bit result.

(a) 16 + 9
(b) 27 + 31
(c) -4 + 19
(d) 3 + -32
(e) -16 + -9
(f) -27 + -31

Solution
The range of 6-bit signed two's complement numbers is (-32,31).

**(a) 16 + 9**

Convert 16 and 9 to unsigned 6-bit binary numbers. Both are positive so we do not need to take the two's complement.

$$
\begin{aligned}
16/16 &= \underline{1}r0 \\
0/8 &= \underline{0}r0 \\
0/4 &= \underline{0}r0 \\
0/2 &= \underline{0}r0 \\
0/1 &= \underline{0}r0
\end{aligned}
$$

$\Rightarrow$ 16 = 01.0000 implicitly sign extending.

$$
\begin{aligned}
9/8 &= \underline{1}r1 \\
1/4 &= \underline{0}r1 \\
1/2 &= \underline{0}r1 \\
1/1 &= \underline{1}r0
\end{aligned}
$$

$\Rightarrow$ 9 = 00.1001 implicitly sign extending.
Adding the two binary numbers

$$
\begin{aligned}
16 + 9 &= 01.0000 + 00.1001 \\
16 + 9 &= 01.1001
\end{aligned}
$$

$\Rightarrow 16 + 9 = 01.1001$ and does not overflow a 6-bit result.

**(b) 27 + 31**

Convert 27 and 31 to unsigned 6-bit binary numbers. Both are positive so we do not need to take the two's complement.

$$
\begin{aligned}
27/16 &= \underline{1}r11 \\
11/8 &= \underline{1}r3 \\
3/4 &= \underline{0}r3 \\
3/2 &= \underline{1}r1 \\
1/1 &= \underline{1}r0
\end{aligned}
$$

$\Rightarrow 27 = 01.1011$ implicitly sign extending.

$$\begin{aligned} 31/16 &= \underline{1}r15 \\ 15/8 &= \underline{1}r7 \\ 7/4 &= \underline{1}r3 \\ 3/2 &= \underline{1}r1 \\ 1/1 &= \underline{1}r0 \end{aligned}$$

$\Rightarrow 31 = 01.1111$ implicitly sign extending.
Adding the two binary numbers

$$\begin{aligned} 27 + 31 &= 01.1011 + 01.1111 \\ 27 + 31 &= 11.1010 \end{aligned}$$

$\Rightarrow$ The numbers being added have the same sign bit and the result 11.1010 has the opposite sign bit. This indicates overflow. Also, the result is outside the range of 6-bit two's complement signed numbers.

**(c) -4 + 19**

Convert the magnitude of -4 and 19 to unsigned 6-bit binary numbers. -4 is negative so we need to take the two's complement. We do not need to take the two's complement of 19.

$$\begin{aligned} 4/4 &= \underline{1}r0 \\ 0/2 &= \underline{0}r0 \\ 0/1 &= \underline{0}r0 \end{aligned}$$

$\Rightarrow 00.0100$ implicitly sign extending.
Take the two's complement by inverting and adding 1:

$$\begin{aligned} \neg\, 00.0100 &= 11.1011 \\ 11.1011 + 00.0001 &= 11.1100 \end{aligned}$$

$$\begin{aligned} 19/16 &= \underline{1}r3 \\ 3/8 &= \underline{0}r3 \\ 3/4 &= \underline{0}r3 \\ 3/2 &= \underline{1}r1 \\ 1/1 &= \underline{1}r0 \end{aligned}$$

$\Rightarrow 01.0011$ implicitly sign extending.
Adding the two binary numbers

$$\begin{aligned} -4 + 19 &= 11.1100 + 01.0011 \\ -4 + 19 &= 00.1111 \end{aligned}$$

$\boxed{\Rightarrow -4+19 = 00.1111 \text{ and does not overflow a 6-bit two's complement signed binary number.}}$

**(d) 3 + -32**

Convert 3 to 6-bit unsigned binary. Convert the magnitude of -32 to 6-bit unsigned binary, then take the two's complement.

By inspection, 3 = 00.0011. Also by inspection, 32 = 10.0000.

Take the two's complement of 10.0000 by inverting and adding 1:

$$\begin{aligned} \neg\, 10.0000 &= 01.1111 \\ 01.1111 + 00.0001 &= 10.0000 \end{aligned}$$

Add the two numbers:

$$\begin{aligned} 3 - 32 &= 00.0011 + 10.0000 \\ 3 - 32 &= 10.0011 \end{aligned}$$

$\boxed{\Rightarrow 3 - 32 = 10.0011 \text{ and does not overflow a 6-bit two's complement signed binary number.}}$

**(e) -16 + -9**

Mix it up a bit in this one. Rather than sign extending the magnitude while unsigned, take two's complement first, then do sign extension of the signed binary number.

Convert the magnitude of $-16$ and $-9$ to unsigned binary. Then take the two's complement. Perform sign extension and then add the 6-bit two's complement signed binary numbers to find the result.

$$\begin{aligned} 16/16 &= \underline{1}r0 \\ 0/8 &= \underline{0}r0 \\ 0/4 &= \underline{0}r0 \\ 0/2 &= \underline{0}r0 \\ 0/1 &= \underline{0}r0 \end{aligned}$$

$\Rightarrow 16 = 1.0000$. No implicit extension.

$$\begin{aligned} 9/8 &= \underline{1}r1 \\ 1/4 &= \underline{0}r1 \\ 1/2 &= \underline{0}r1 \\ 1/1 &= \underline{1}r0 \end{aligned}$$

$\Rightarrow 9 = 1001$. No implicit extension.

Take the two's complement of 1.0000 (16) by inverting and adding 1:

$$\begin{aligned} \neg\, 1.0000 &= 0.1111 \\ 0.1111 + 0.0001 &= 1.0000 \end{aligned}$$

Take two's complement of 1001 (9) by inverting and adding 1:

$$\neg\, 1001 = 0110$$
$$0110 + 0001 = 0111$$

Sign extend both terms:

$$-16 = 11.0000$$
$$-9 = 11.0111$$

Add both terms:

$$-16 + -9 = 11.0000 + 11.0111$$
$$-16 + -9 = 10.0111$$

$\Rightarrow -16 + -9 = 10.0111$ and does not overflow a 6-bit two's complement signed binary number.

**(f) -27 + -31**

Convert the magnitude of $-27$ and $-31$ to unsigned binary. Then take the two's complement. Perform sign extension and then add the 6-bit two's complement signed binary numbers to find the result.

Convert magnitude 27 to unsigned binary.

$$27/16 = \underline{1}r11$$
$$11/8 = \underline{1}r3$$
$$3/4 = \underline{0}r3$$
$$3/2 = \underline{1}r1$$
$$1/1 = \underline{1}r0$$

$\Rightarrow 27 = 1.1011$. No implicit extension.

Convert magnitude 31 to unsigned binary.

$$31/16 = \underline{1}r15$$
$$15/8 = \underline{1}r7$$
$$7/4 = \underline{1}r3$$
$$3/2 = \underline{1}r1$$
$$1/1 = \underline{1}r0$$

$\Rightarrow 31 = 1.1111$. No implicit extension.

Take two's complement of 1.1011 (27) by inverting and adding 1:

$$\neg\, 1.1011 = 00100$$
$$0.0100 + 0.0001 = 0.0101$$

Take two's complement of 1.1111 (31) by inverting and adding 1:

$$\neg\, 1.1111 = 0.0000$$
$$0.0000 + 0.0001 = 0.0001$$

Sign extend both terms.

$$-27 = 10.0101$$
$$-31 = 10.0001$$

$$-27 + -31 = 10.0101 + 10.0001$$
$$-27 + -31 = 00.0110$$

From the text, page 18:

> Adding a positive number to a negative number never causes an overflow. Unlike unsigned numbers, a carry out of the most significant column does not indicate overflow. Instead, *overflow occurs if the two numbers being added have the same sign bit and the result has the opposite sign bit.*

$\Rightarrow -27 + -31 = 00.0110$ and the result overflows for the reason cited above.

$\bowtie$

EXERCISE 1.58 Perform the following additions of unsigned hexadecimal numbers. Indicate whether or not the sum overflows an 8-bit (two hex digit) result.

(a) 0x7 + 0x9
(b) 0x13 + 0x28
(c) 0xAB + 0x3E
(d) 0x8F + 0xAD

SOLUTION

**(a) 0x7 + 0x9**

By inspection, convert 0x7 and 0x9 to binary. Since each hexadecimal digit is represented by a 4-bit binary number, we can convert each number to it's 4-bit binary representation and substitute.

$$0x7 = 0111$$
$$0x9 = 1001$$

Adding the two numbers yields

$$0x7 + 0x9 = 0111 + 1001$$
$$0x7 + 0x9 = 1.0000$$

$\Rightarrow$ 0x7 + 0x9 = 0x10 (1.0000) and does not overflow an 8-bit result.

**(b) 0x13 + 0x28**

By inspection, convert 0x13 and 0x28 to binary. Since each hexadecimal digit is represented by a 4-bit binary number, we can convert each number to it's 4-bit binary representation and substitute.

$$0\text{x}13 = 0001.0011$$
$$0\text{x}28 = 0010.1000$$

Adding the two numbers yields

$$0\text{x}13 + 0\text{x}28 = 0001.0011 + 0010.1000$$
$$0\text{x}13 + 0\text{x}28 = 0011.1011$$

Converting the binary result back to hexadecimal

$$0011.1011 = 0\text{x}3.0\text{xB}$$
$$0011.1011 = 0\text{x}3\text{B}$$

$\Rightarrow$ 0x13 + 0x28 = 0x3B (0011.1011) and does not overflow an 8-bit result.

**(c) 0xAB + 0x3E**

By inspection, convert 0xAB and 0x28 to binary. Since each hexadecimal digit is represented by a 4-bit binary number, we can convert each number to it's 4-bit binary representation and substitute.

$$0\text{xAB} = 1010.1011$$
$$0\text{x}3\text{E} = 0011.1110$$

Adding the two numbers yields

$$0\text{xAB} + 0\text{x}3\text{E} = 1010.1011 + 0011.1110$$
$$0\text{xAB} + 0\text{x}3\text{E} = 1110.1001$$

Converting the binary result back to hexadecimal

$$1110.1001 = 0\text{xE}.0\text{x}9$$
$$1110.1001 = 0\text{xE}9$$

$\Rightarrow$ 0xAB + 0x3E = 0xE9 (1110.1001) and does not overflow an 8-bit result.

**(d) 0x8F + 0xAD**

By inspection, convert 0x8F and 0xAD to binary. Since each hexadecimal digit is represented by a 4-bit binary number, we can convert each number to it's 4-bit binary representation and substitute.

$$\begin{aligned}0\text{x8F} &= 1000.1111\\0\text{xAD} &= 1010.1101\end{aligned}$$

Adding the two numbers yields

$$\begin{aligned}0\text{x8F} + 0\text{xAD} &= 1000.1111 + 1010.1101\\0\text{x8F} + 0\text{xAD} &= 0001.0011.1100\end{aligned}$$

Converting the binary result back to hexadecimal

$$\begin{aligned}0001.0011.1100 &= 0\text{x1}.0\text{x3}.0\text{xC}\\0001.0011.1100 &= 0\text{x13C}\end{aligned}$$

$\Rightarrow$ 0x8F + 0xAD = 0x13C (0001.0011.1100) and does overflow an 8-bit result.

$\bowtie$

EXERCISE 1.60 Convert the following decimal numbers to 5-bit two's complement binary numbers and subtract them. Indicate whether or not the difference overflows a 5-bit result.

(a) 9 - 7
(b) 12 - 15
(c) -6 - 11
(d) 4 - -8

SOLUTION

Subtraction is performed by taking the two' complement and adding.

Overflow works differently for signed numbers than unsigned numbers. From page 18 of the text:

> Adding a positive number to a negative number never causes an overflow. Unlike unsigned numbers, a carry out of the most significant column does not indicate overflow. Instead, *overflow occurs if the two numbers being added have the same sign bit and the result has the opposite sign bit.*

**(a) 9 - 7**

By inspection, 9 is binary 0.1001 and 7 is binary 0.0111. Subtraction is accomplished by taking the two's complement and adding. We do not need to take the two's complement of 9 since it is already a valid signed number.

Take the two's complement of 7 by inverting and adding 1. Inverting yields $\Rightarrow$ 1.1000. Adding 1 yields $\Rightarrow$ 1.1001.

Now add the two numbers using normal binary addition, carrying as necessary:

$$9 - 7 = 0.1001 + 1.1001$$
$$9 - 7 = 0.0010$$

There was a carry out of the *most significant column* but that does not indicate an overflow. It is not possible to overflow by adding a positive and a negative number. Therefore, the result is:

$\Rightarrow$ $9 - 7 = 0.0010$, or 2 and does not overflow a 5-bit result.

**(b) 12 - 15**

By inspection, 12 is binary 0.1100 and 15 is binary 0.1111. Subtraction is accomplished by taking the two's complement and adding. We do not need to take the two's complement of 12 since it is already a valid signed number.

Take the two's complement of 15 by inverting and adding 1. Inverting yields $\Rightarrow$ 1.0000. Adding 1 yields $\Rightarrow$ 1.0001.

Now add the two numbers using normal binary addition, carrying as necessary:

$$12 - 15 = 0.1100 + 1.0001$$
$$12 - 15 = 1.1101$$

The result is:

$\Rightarrow$ $12 - 15 = 1.1101$, or -3 and does not overflow a 5-bit result.

**(c) -6 - 11**

By inspection, the 5-bit magnitude of -6 and -11 is 0.0110 and 0.1011, respectively. Since both are negative, we must take the two's complement of each.

Invert both:

$$\neg\, 0.0110 = 1.1001$$
$$\neg\, 0.1011 = 1.0100$$

Adding 1 yields:

$$1.1001 + 0.0001 = 1.1010$$
$$1.0100 + 0.0001 = 1.0101$$

Now add both numbers:

$$-6 - 11 = 1.1010 + 1.0101$$
$$-6 - 11 = 0.1111$$

There was a carry out of the *most significant digit*. Both numbers being added have the same sign bit and the result has an opposite sign bit. Therefore, the result overflows.

**(d) 4 - -8**

By inspection, the 5-bit magnitude of 4 and -8 is 0.0100 and, 0.1000 respectively. Since we are subtracting a negative number, the net result is positive. Therefore we do not need to take the two's complement of either number.

Now add both numbers:

$$0.0100 + 0.1000 = 0.1100$$

$\Rightarrow 4 - -8 = 0.1100$, or 12 and does not overflow a 5-bit result.

⋈

EXERCISE 1.65 Answer the following questions related to BCD systems (see EXERCISE 1.64 for a definition of BCD).

(a) Write 371 in BCD
(b) Convert $000110000111_{BCD}$ to decimal
(c) Convert $10010101_{BCD}$ to binary
(d) Explain the disadvantages of BCD when compared to binary representations of numbers

SOLUTION

**(a) Write 371 in BCD**

Each decimal digit is represented by it's 4-digit binary equivalent. By inspection, we can write:

$$3 = 0011$$
$$7 = 0111$$
$$1 = 0001$$

Combining the binary representations in the proper order yields the BCD result:

$\Rightarrow 371 = 0011.0111.0001_{BCD}$

**(b) Convert $000110000111_{BCD}$ to decimal**

Each 4-bit grouping in the 12-bit BCD number represents a decimal digit. We can more clearly write the number as $0001.1000.0111_{BCD}$. By inspection, we can write:

$$0001 = 1$$
$$1000 = 8$$
$$0111 = 7$$

Combining the BCD representations in proper order yields the decimal result:

$\Rightarrow 0001.1000.0111_{BCD} = 187$

**(c) Convert $10010101_{BCD}$ to binary**

The BCD number must first be converted to decimal. Then it can be converted to binary.

Each 4-bit grouping in the 8-bit BCD number represents a decimal digit. By inspection, we can write:

$$1001 = 9$$
$$0101 = 5$$

Therefore, the BCD number corresponds to 95. This may be converted to binary by using method 1 demonstrated in Example 1.5.

$$\begin{aligned} 95/64 &= \underline{1}r31 \\ 31/32 &= \underline{0}r31 \\ 31/16 &= \underline{1}r15 \\ 15/8 &= \underline{1}r7 \\ 7/4 &= \underline{1}r3 \\ 3/2 &= \underline{1}r1 \\ 1/1 &= \underline{1}r1 \end{aligned}$$

Assemble the quotients in *significant* digit order to yield the result:

$\Rightarrow 10010101_{BCD} = 101.1111$

**(d) Explain the disadvantages of BCD when compared to binary representations of numbers**

Addition of BCD numbers doesn't work directly. Also, the representation doesn't maximize the amount of information that can be stored; for example 2 BCD digits requires 8 bits and can store up to 100 values (0-99) - unsigned 8- bit binary can store 28 (256) values.