Rules of Binary Addition

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0 = 1
- 1 + 1 = 0, and *carry* 1 to the next more significant bit

For example,

00011010 + 00001100 = 00100110

1 1		*carries*
0 0 0 1 1 0 1 0	=	$26_{(\text{base } 10)}$
+ 0 0 0 0 1 1 0 0	=	$12_{(\text{base } 10)}$
0 0 1 0 0 1 1 0	=	$38_{(\text{base } 10)}$

00010011 + 00111110 = 01010001

1 1 1 1 1		*carries*
0 0 0 1 0 0 1 1	=	$19_{(\text{base } 10)}$
+ 0 0 1 1 1 1 1 0	=	$62_{(\text{base } 10)}$
0 1 0 1 0 0 0 1	=	$81_{(\text{base } 10)}$

Rules of Binary Subtraction

- 0 - 0 = 0
- 0 - 1 = 1, and *borrow* 1 from the next more significant bit
- 1 - 0 = 1
- 1 - 1 = 0

For example,

00100101 - 00010001 = 00010100

0		*borrows*
0 0 ~~1~~ ^{1}0 0 1 0 1	=	$37_{(\text{base } 10)}$
- 0 0 0 1 0 0 0 1	=	$17_{(\text{base } 10)}$
0 0 0 1 0 1 0 0	=	$20_{(\text{base } 10)}$

00110011 - 00010110 = 00011101

0 ^{1}0 1		*borrows*
0 0 ~~1~~ ~~1~~ ~~0~~ ^{1}0 1 1	=	$51_{(\text{base } 10)}$
- 0 0 0 1 0 1 1 0	=	$22_{(\text{base } 10)}$
0 0 0 1 1 1 0 1	=	$29_{(\text{base } 10)}$

Rules of Binary Multiplication

- 0 x 0 = 0
- 0 x 1 = 0
- 1 x 0 = 0
- 1 x 1 = 1, and no carry or borrow bits

For example,

00101001 × 00000110 = 11110110

$$\begin{array}{rcl}
0\ 0\ 1\ 0\ 1\ 0\ 0\ 1 & = & 41_{(\text{base } 10)} \\
\times\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0 & = & 6_{(\text{base } 10)} \\
\hline
0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 & & \\
0\ 0\ 1\ 0\ 1\ 0\ 0\ 1 & & \\
0\ 0\ 1\ 0\ 1\ 0\ 0\ 1 & & \\
\hline
0\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0 & = & 246_{(\text{base } 10)}
\end{array}$$

00010111 × 00000011 = 01000101

$$\begin{array}{rcl}
0\ 0\ 0\ 1\ 0\ 1\ 1\ 1 & = & 23_{(\text{base } 10)} \\
\times\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1 & = & 3_{(\text{base } 10)} \\
\hline
\mathit{1\ 1\ 1\ 1\ 1} & & \textit{carries} \\
0\ 0\ 0\ 1\ 0\ 1\ 1\ 1 & & \\
0\ 0\ 0\ 1\ 0\ 1\ 1\ 1 & & \\
\hline
0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 1 & = & 69_{(\text{base } 10)}
\end{array}$$

Another Method: Binary multiplication is the same as repeated binary addition; add the multicand to itself the multiplier number of times.

For example,

00001000 × 00000011 = 00011000

$$\begin{array}{rcl}
\mathit{1} & & \textit{carries} \\
0\ 0\ 0\ 0\ 1\ 0\ 0\ 0 & = & 8_{(\text{base } 10)} \\
0\ 0\ 0\ 0\ 1\ 0\ 0\ 0 & = & 8_{(\text{base } 10)} \\
+\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0 & = & 8_{(\text{base } 10)} \\
\hline
0\ 0\ 0\ 1\ 1\ 0\ 0\ 0 & = & 24_{(\text{base } 10)}
\end{array}$$

Binary Division

Binary division is the repeated process of subtraction, just as in decimal division.

For example,

00101010 ÷ 00000110 = 00000111

$$
\begin{array}{rl}
 & 1\ 1\ 1 = 7_{(base\ 10)} \\
\hline
1\ 1\ 0\ ) & 0\ 0\ \cancel{1}\ {}^{1}0\ 1\ 0\ 1\ 0 = 42_{(base\ 10)} \\
 & -\ 1\ 1\ 0 = 6_{(base\ 10)} \\
\hline
 & \mathit{1} \quad \textit{borrows} \\
 & \cancel{1}\ \cancel{0}\ {}^{1}0\ 1 \\
 & -\ 1\ 1\ 0 \\
\hline
 & 1\ 1\ 0 \\
 & -\ 1\ 1\ 0 \\
\hline
 & 0
\end{array}
$$

10000111 ÷ 00000101 = 00011011

$$
\begin{array}{rl}
 & 1\ 1\ 0\ 1\ 1 = 27_{(base\ 10)} \\
\hline
1\ 0\ 1\ ) & \cancel{1}\ \cancel{0}\ \cancel{0}\ {}^{1}0\ 0\ 1\ 1\ 1 = 135_{(base\ 10)} \\
 & -\ 1\ 0\ 1 = 5_{(base\ 10)} \\
\hline
 & 1\ \cancel{1}\ {}^{1}0 \\
 & -\ 1\ 0\ 1 \\
\hline
 & 1\ 1 \\
 & -\ 0 \\
\hline
 & 1\ 1\ 1 \\
 & -\ 1\ 0\ 1 \\
\hline
 & 1\ 0\ 1 \\
 & -\ 1\ 0\ 1 \\
\hline
 & 0
\end{array}
$$

Notes

Binary Number System

System Digits: 0 and 1

Bit (short for *b*inary dig*it*): A single binary digit

LSB (least significant bit): The rightmost bit

MSB (most significant bit): The leftmost bit

Upper Byte (or nybble): The right-hand byte (or nybble) of a pair

Lower Byte (or nybble): The left-hand byte (or nybble) of a pair

Binary Equivalents

1 Nybble (or nibble) = 4 bits

1 Byte = 2 nybbles = 8 bits

1 Kilobyte (KB) = 1024 bytes

1 Megabyte (MB) = 1024 kilobytes = 1,048,576 bytes

1 Gigabyte (GB) = 1024 megabytes = 1,073,741,824 bytes