Spring 2019

CSE 112 (Computer Fundamentals

Topic: Computer Arithmetic

Department of Computer Science and Engineering
Daffodil International University

References

- *Computer Fundamentals by Pradeep K. Sinha, 6th Edition. [Chapter 5]*

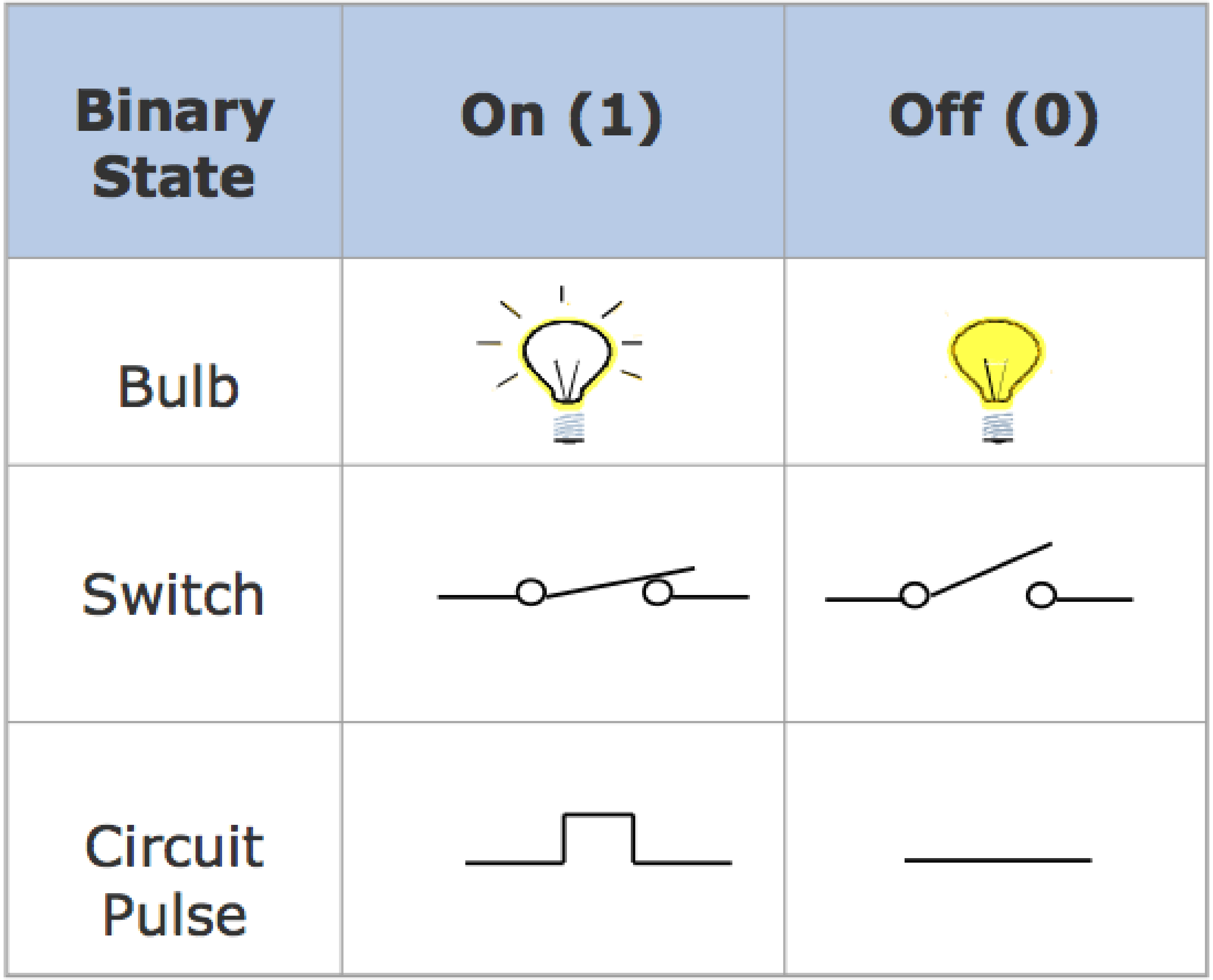
- *Computer Fundamentals and ICT by M. Lutfar Rahman , M. Shamim Kaiser , M. Ariful Rahman , M. Alamgir Hossain.*

  [Chapter 2]

Binary over Decimal

- Information is handled in a computer by electronic/ electrical components
- Electronic components operate in binary mode (can only indicate two states – ON (1) or OFF (0)
- Binary number system has only two digits (0 and 1), and is suitable for expressing two possible states
- In binary system, computer circuits only have to handle two binary digits rather than ten decimal digits causing:
  - Simpler internal circuit design
  - Less expensive
  - More reliable circuits
- Arithmetic rules/processes possible with binary numbers

Examples of a Few Devices That Work in Binary Mode Binary

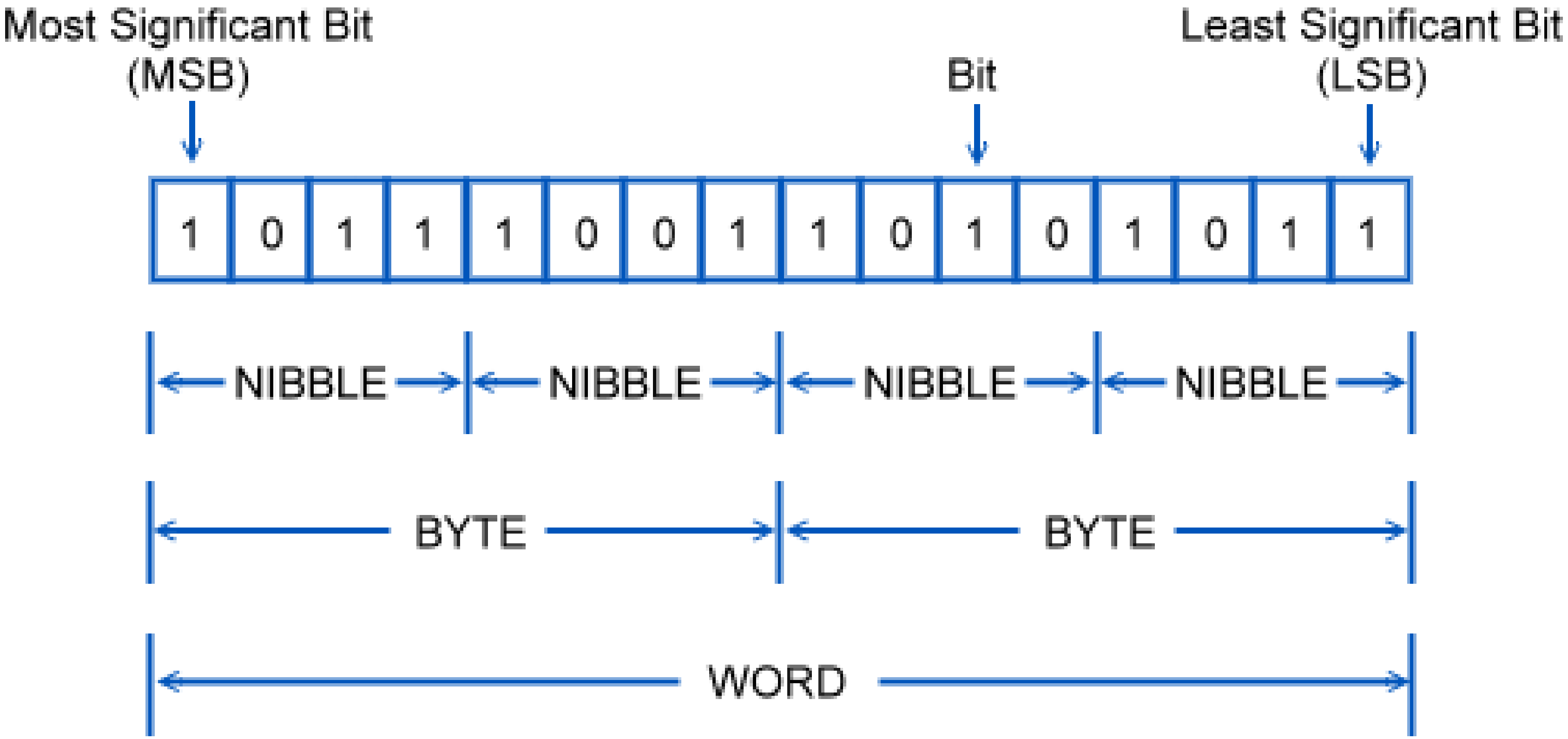


Binary State	On (1)	Off (0)
Bulb		
Switch		
Circuit Pulse		

Binary Number System

- System Digits: 0 and 1
- Bit (short for binary digit): 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
- The term **nibble** used for 4 bits being a subset of byte.

Binary Number System



Most Significant Bit (MSB)

Bit

Least Significant Bit (LSB)

1	0	1	1	1	0	0	1	1	0	1	0	1	0	1	1

NIBBLE NIBBLE NIBBLE NIBBLE

BYTE BYTE

WORD

Binary Equivalents

- 1 Nybble (or nibble) = 4 bits
- 1 Byte = 2 nybbles = 8 bits
- 1 Kilobyte (KB) = 2^{10} or 1024 bytes
- 1 Megabyte (MB) = 2^{10} or 1024 kilobytes = 2^{20} or 1,048,576 bytes
- 1 Gigabyte (GB) = 2^{10} or 1024 megabytes = 2^{30} or 1,073,741,824 bytes

Binary Arithmetic

- Binary arithmetic is simple to learn as binary number system has only two digits – 0 and 1
- Following slides show rules and example for the four basic arithmetic operations using binary numbers

Binary Addition

- Rule for binary addition is as follows:

① 0 + 0 = 0

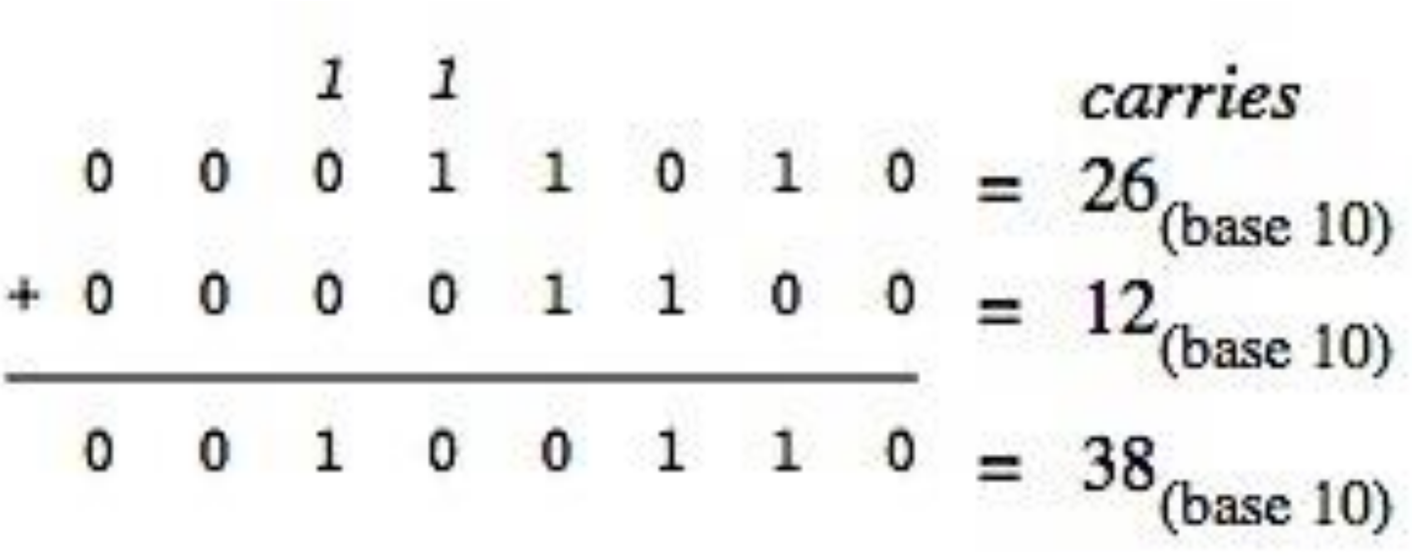
② 0 + 1 = 1

③ 1 + 0 = 1

④ 1 + 1 = 0 plus a carry of 1 to next higher column

Binary Addition

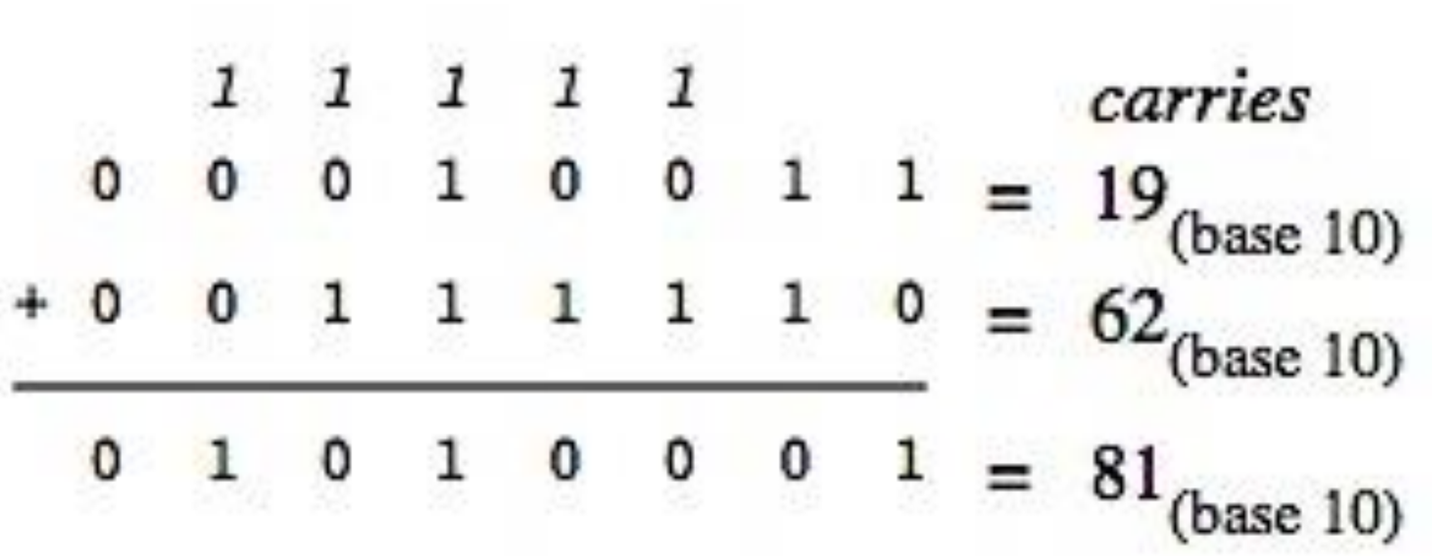
- **Example 1:** $00011010_2 + 00001100_2 = 00100110_2$



$$\begin{array}{cccccccccccl} & & & \mathit{1} & \mathit{1} & & & & & & & \textit{carries}\\ & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & = & & 26_{(\text{base } 10)}\\ + & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & = & & 12_{(\text{base } 10)}\\ \hline & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & = & & 38_{(\text{base } 10)}\end{array}$$

Binary Addition

- **Example 2: $00010011_2 + 00111110_2 = 01010001_2$**



		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	=	$81_{(\text{base } 10)}$

Binary Addition (Example 3)

Example

Add binary numbers 100111 and 11011 in both decimal and binary form

Solution

	Binary	Decimal
carry	11111	1
	100111	39
	+11011	+27
	1000010	66

The addition of three 1s can be broken up into two steps. First, we add only two 1s giving 10 (1 + 1 = 10). The third 1 is now added to this result to obtain 11 (a 1 sum with a 1 carry). Hence, 1 + 1 + 1 = 1, plus a carry of 1 to next higher column.

Binary Subtraction

- Rule for binary subtraction is as follows:

  ① 0 - 0 = 0

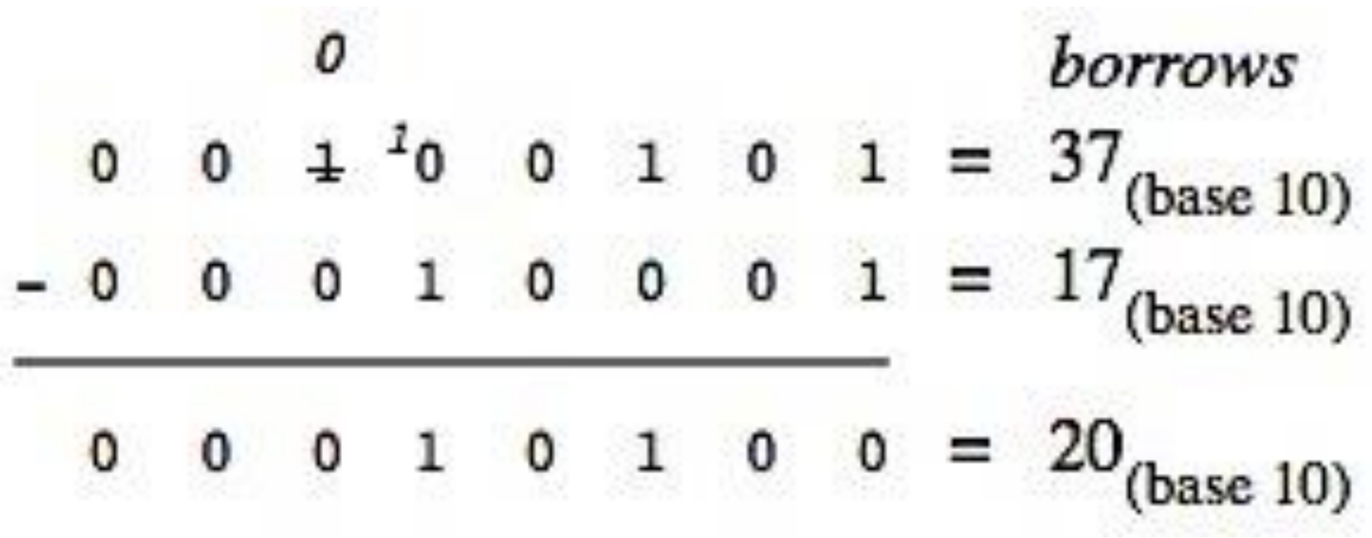
  ② 0 - 1 = 1 with a borrow from the next column

  ③ 1 - 0 = 1

  ④ 1 - 1 = 0

Binary Subtraction

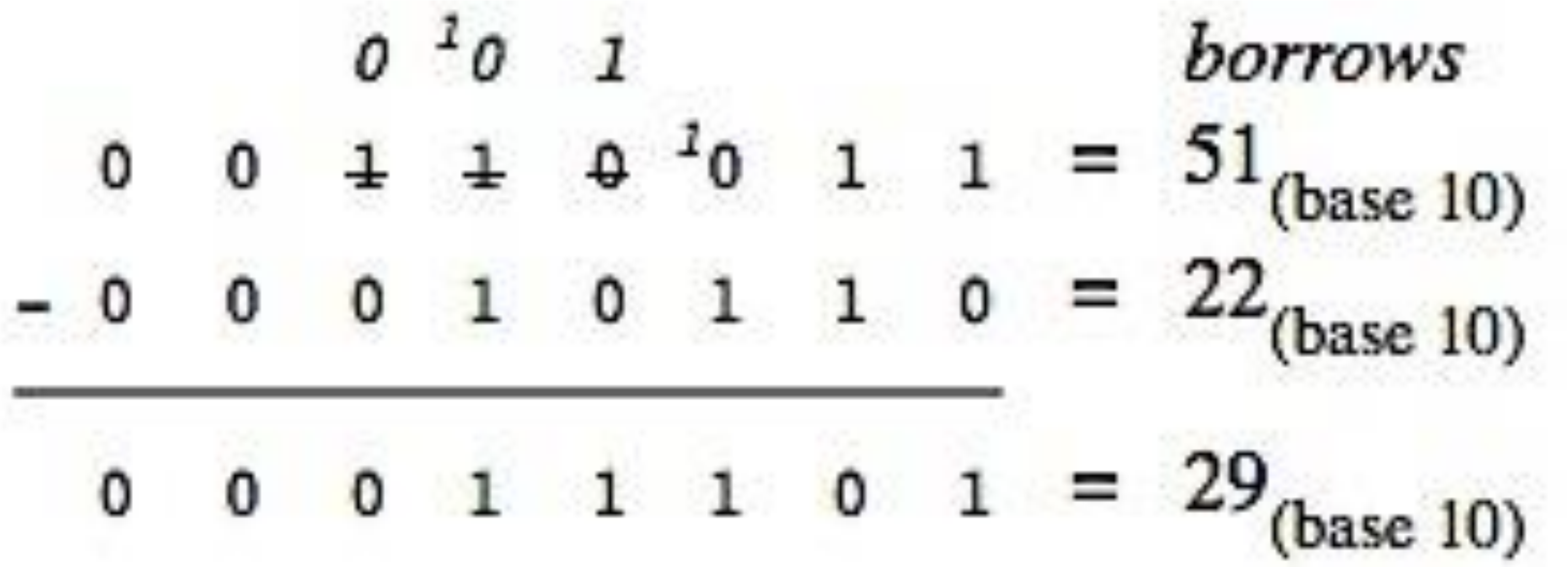
- **Example 1: 00100101_2 - 00010001_2 = 00010100_2**



$$\begin{array}{cccccccccl}
 & & & 0 & & & & & & \textit{borrows} \\
 & 0 & 0 & \cancel{1} & {}^{1}0 & 0 & 1 & 0 & 1 & = 37_{(\text{base } 10)} \\
- & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & = 17_{(\text{base } 10)} \\
\hline
 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & = 20_{(\text{base } 10)}
\end{array}$$

Binary Subtraction

- **Example 2: 00110011_2 - 00010110_2 = 00011101_2**



$$
\begin{array}{ccccccccccl}
 & & & 0 & {}^{1}0 & 1 & & & & & \textit{borrows} \\
 & 0 & 0 & \cancel{1} & \cancel{1} & \cancel{0} & {}^{1}0 & 1 & 1 & = & 51_{(\text{base }10)} \\
- & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & = & 22_{(\text{base }10)} \\
\hline
 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & = & 29_{(\text{base }10)}
\end{array}
$$

Binary Multiplication

- Table for binary multiplication is as follows:

① 0 x 0 = 0

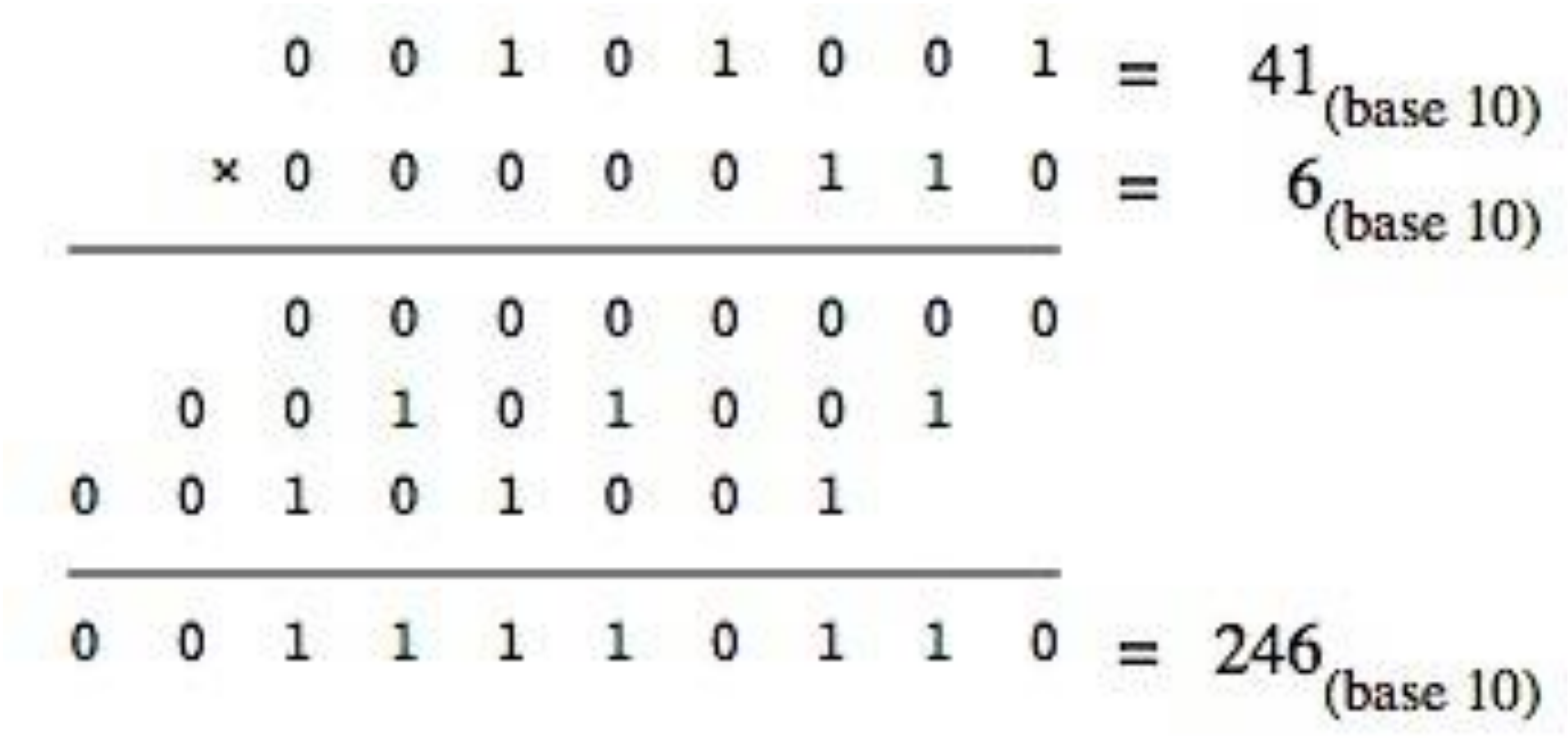
② 0 x 1 = 0

③ 1 x 0 = 0

④ 1 x 1 = 1

Binary Multiplication

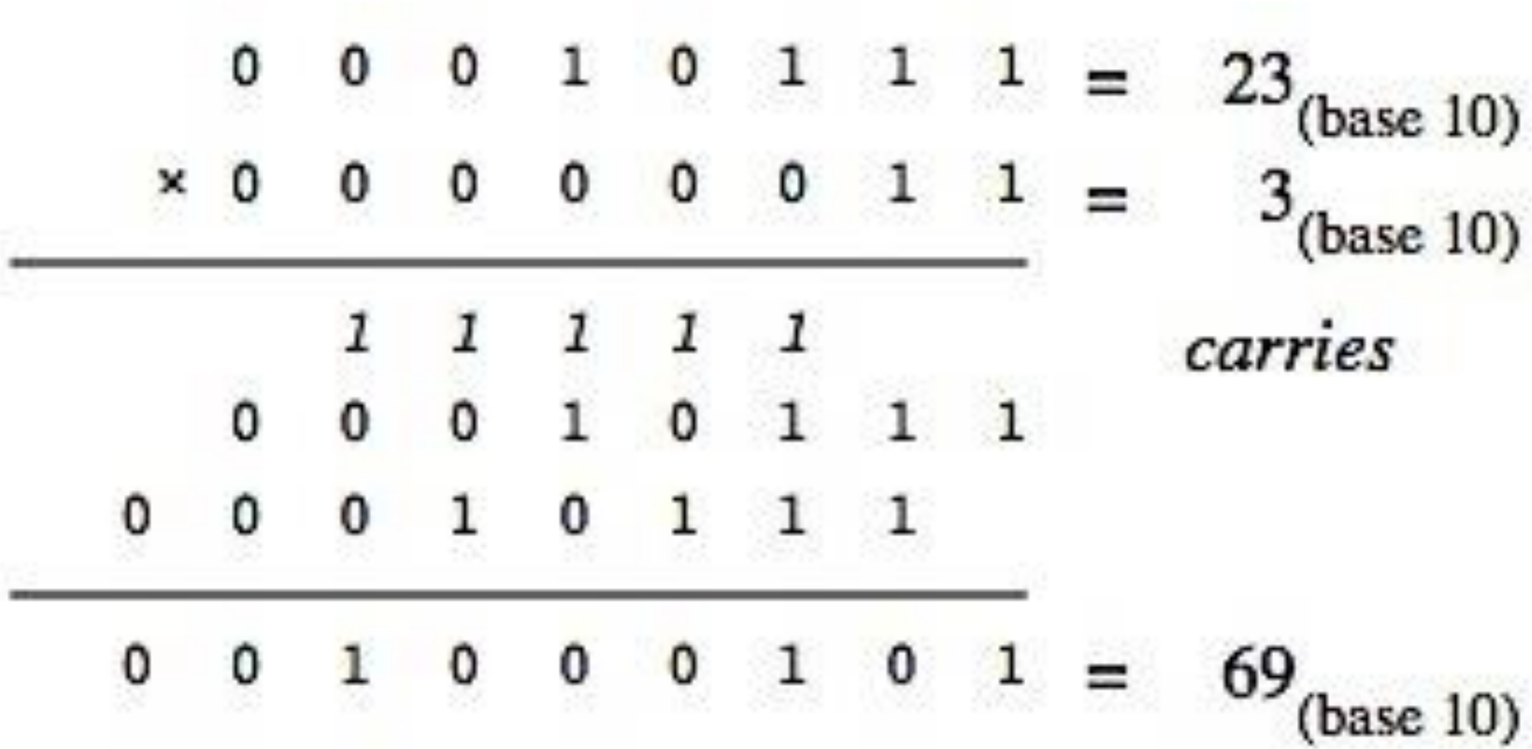
- **Example 1: $00101001_2 \times 00000110_2 = 11110110_2$**



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

Binary Multiplication

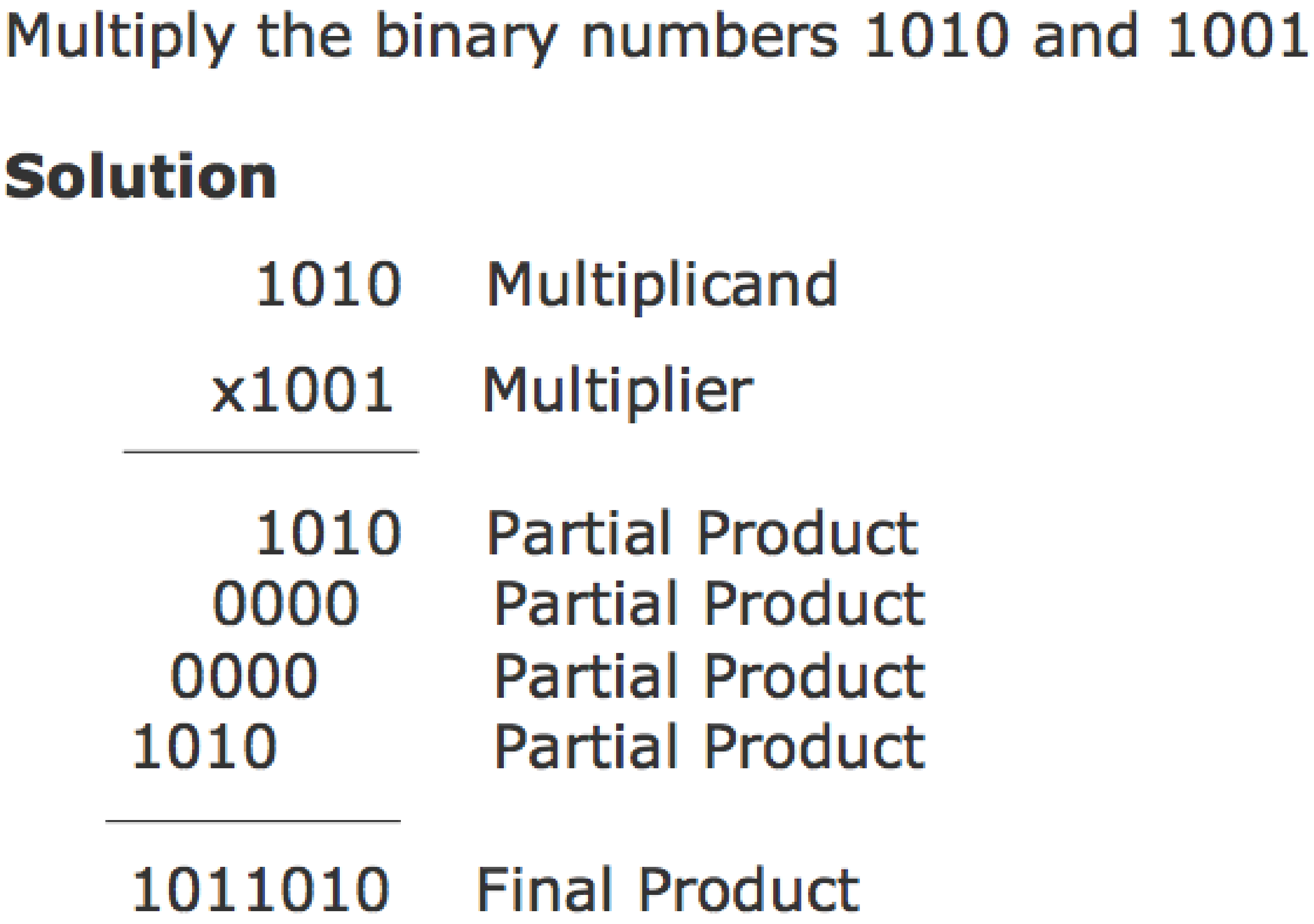
- Example 2: $00010111_2 \times 00000011_2 = 01000101_2$



```
      0  0  0  1  0  1  1  1  =  23(base 10)
    × 0  0  0  0  0  0  1  1  =   3(base 10)
  ------------------------------
         1  1  1  1  1           carries
      0  0  0  1  0  1  1  1
   0  0  0  1  0  1  1  1
  ------------------------------
   0  0  1  0  0  0  1  0  1  =  69(base 10)
```

Binary Multiplication

- **Example 3:**



Multiply the binary numbers 1010 and 1001

Solution

```
      1010     Multiplicand
     x1001     Multiplier
  ------------
      1010     Partial Product
     0000      Partial Product
    0000       Partial Product
   1010        Partial Product
  ------------
   1011010     Final Product
```

Binary Division

- Table for binary division is as follows:
  1. $0 \div 0 =$ Divide by zero error
  2. $0 \div 1 = 0$
  3. $1 \div 0 =$ Divide by zero error
  4. $1 \div 1 = 1$
- As in the decimal number system (or in any other number system), division by zero is meaningless
- The computer deals with this problem by raising an error condition called 'Divide by zero' error

Rules for Binary Division

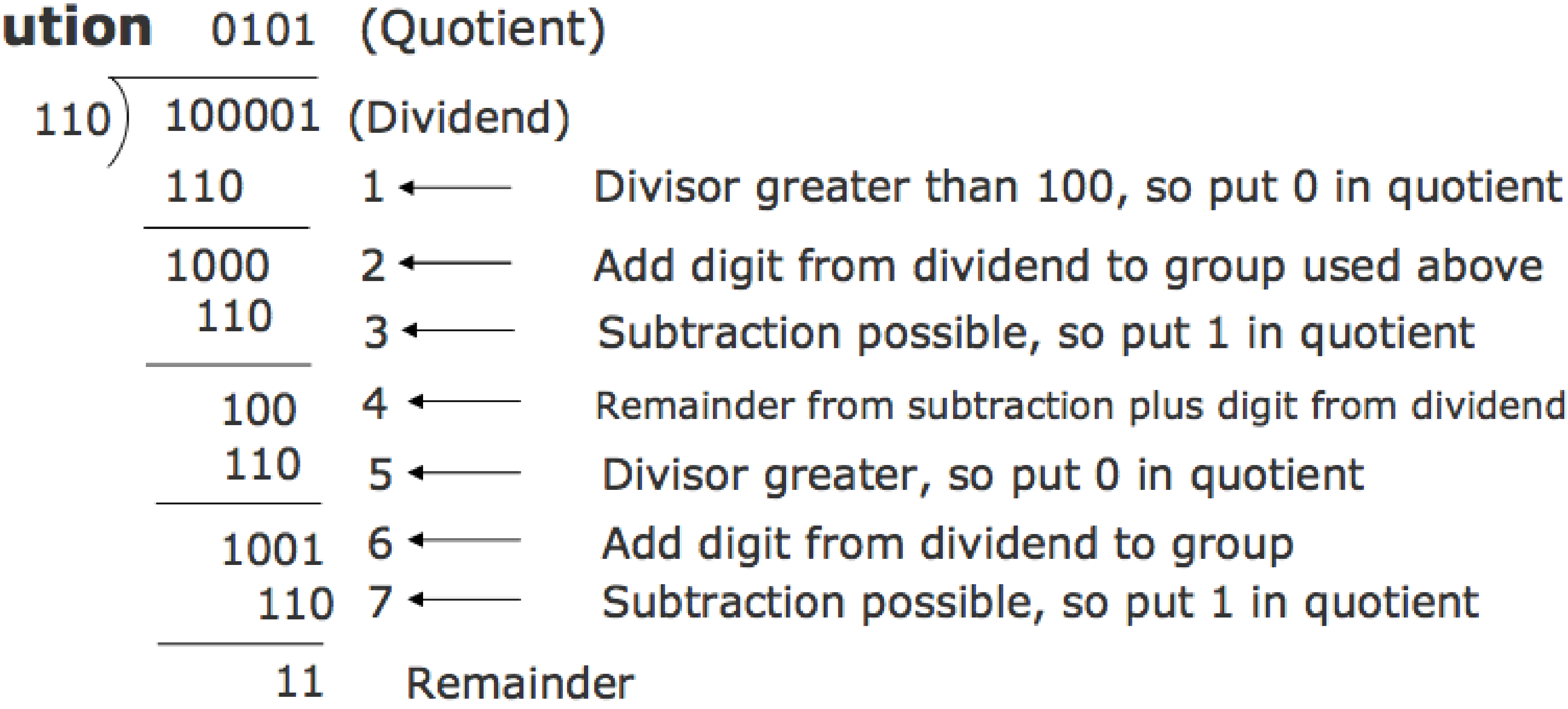
1. Start from the left of the dividend
2. Perform a series of subtractions in which the divisor is subtracted from the dividend
3. If subtraction is possible, put a 1 in the quotient and subtract the divisor from the corresponding digits of dividend
4. If subtraction is not possible (divisor greater than remainder), record a 0 in the quotient
5. Bring down the next digit to add to the remainder digits. Proceed as before in a manner similar to long division

Binary Division (Example 1)

Example

Divide 100001_2 by 110_2

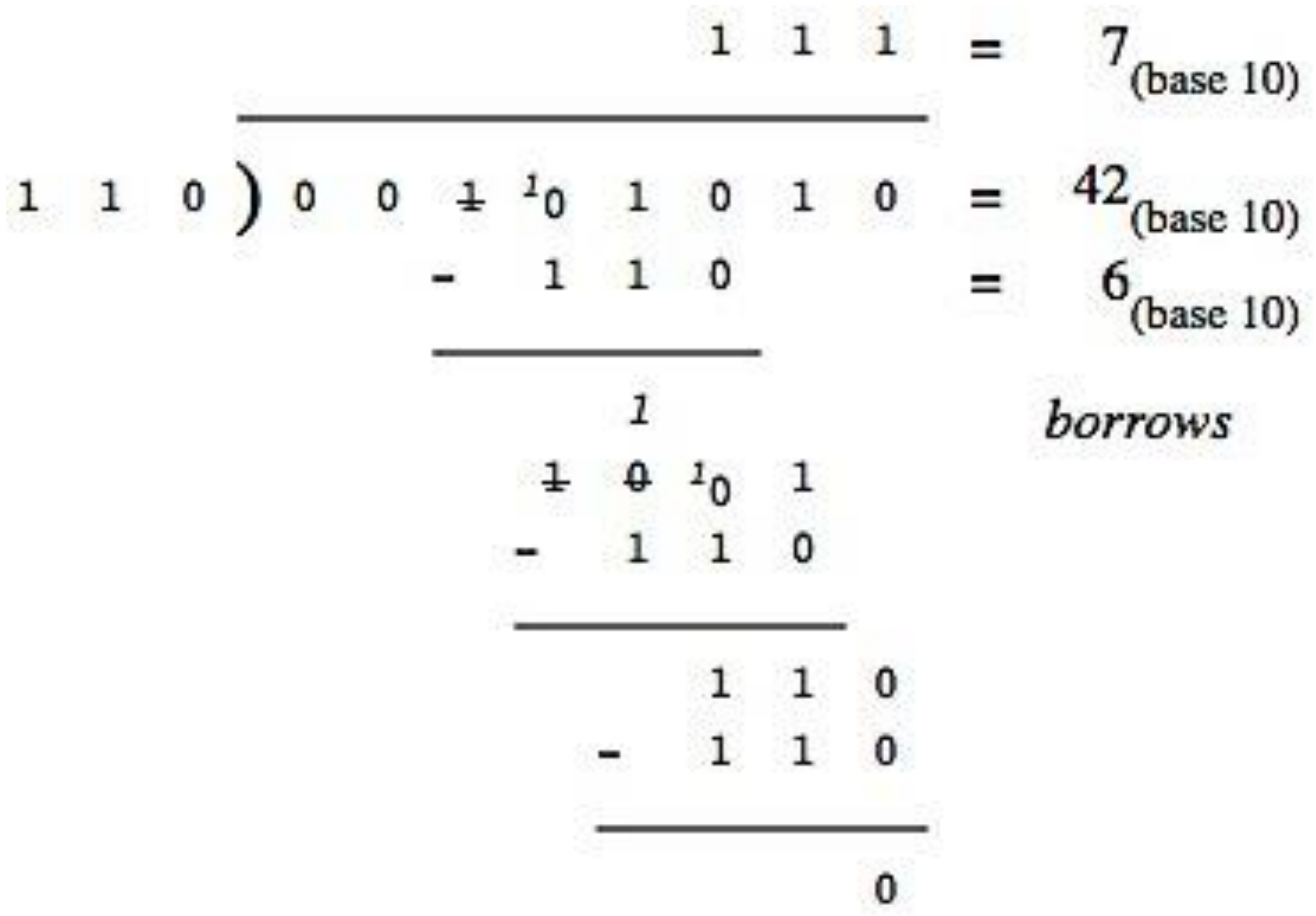
Solution



```
        0101    (Quotient)
110 ) 100001   (Dividend)
      110       1 ←   Divisor greater than 100, so put 0 in quotient
      ----
      1000      2 ←   Add digit from dividend to group used above
       110      3 ←   Subtraction possible, so put 1 in quotient
      ----
        100     4 ←   Remainder from subtraction plus digit from dividend
        110     5 ←   Divisor greater, so put 0 in quotient
       -----
        1001    6 ←   Add digit from dividend to group
         110    7 ←   Subtraction possible, so put 1 in quotient
       -----
          11      Remainder
```

Binary Division (Example 2)

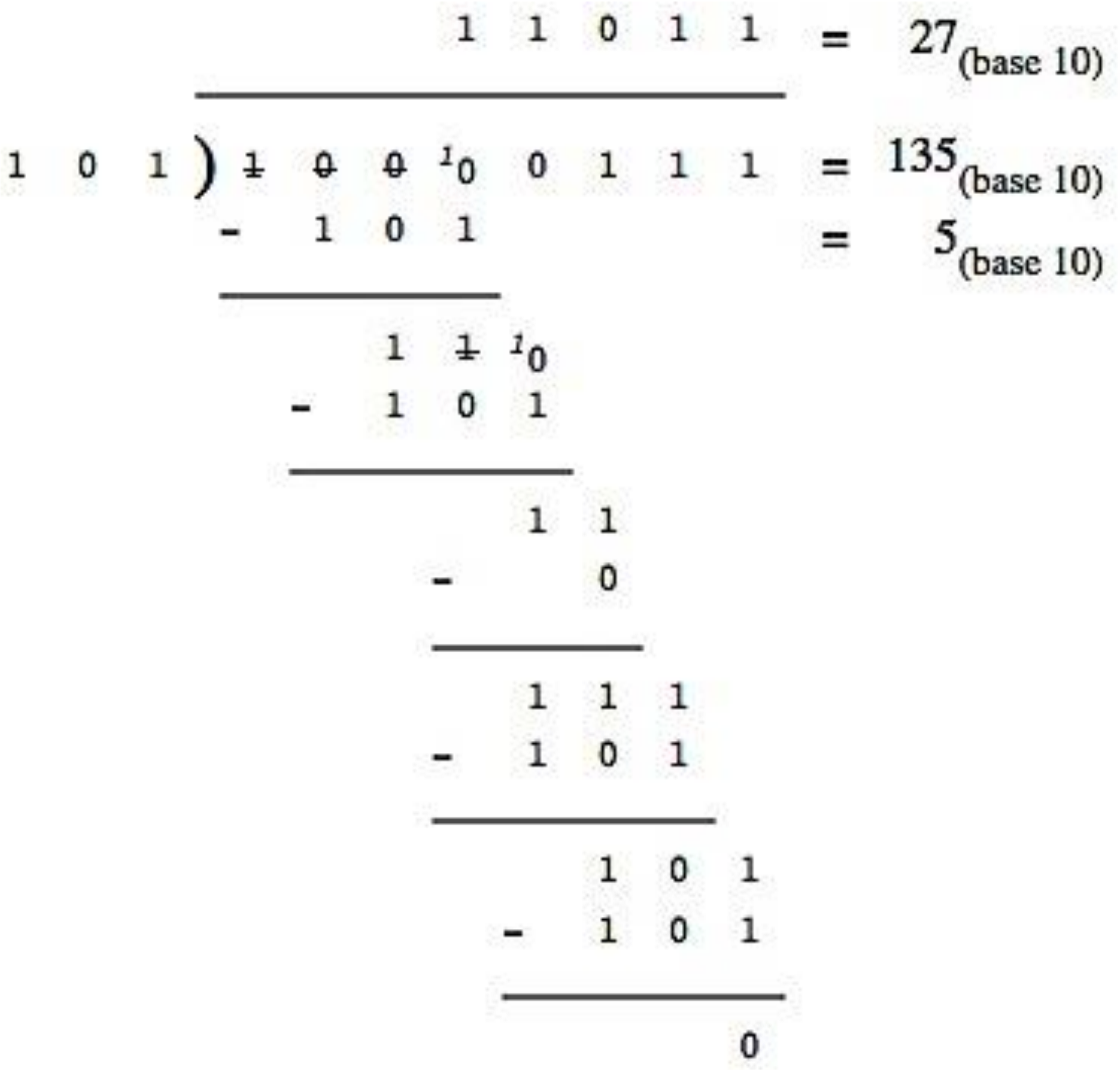
- **Example: $00101010_2 \div 00000110_2 = 00000111_2$**



$$\begin{array}{ccccccccccccl}
 & & & & & & & & & 1 & 1 & 1 & = 7_{(\text{base } 10)} \\ \cline{5-12}
1 & 1 & 0 & \big) & 0 & 0 & \cancel{1} & {}^{1}0 & 1 & 0 & 1 & 0 & = 42_{(\text{base } 10)} \\
 & & & & & & - & 1 & 1 & 0 & & & = 6_{(\text{base } 10)} \\ \cline{7-10}
 & & & & & & & & \mathit{1} & & & & \textit{borrows} \\
 & & & & & & & \cancel{1} & \cancel{0} & {}^{1}0 & 1 & & \\
 & & & & & & & - & 1 & 1 & 0 & & \\ \cline{8-12}
 & & & & & & & & & 1 & 1 & 0 & \\
 & & & & & & & & - & 1 & 1 & 0 & \\ \cline{9-12}
 & & & & & & & & & & & 0 &
\end{array}$$

Binary Division (Example 3)

- Example: $10000111_2 \div 00000101_2 = 00011011_2$



The End