本节主题

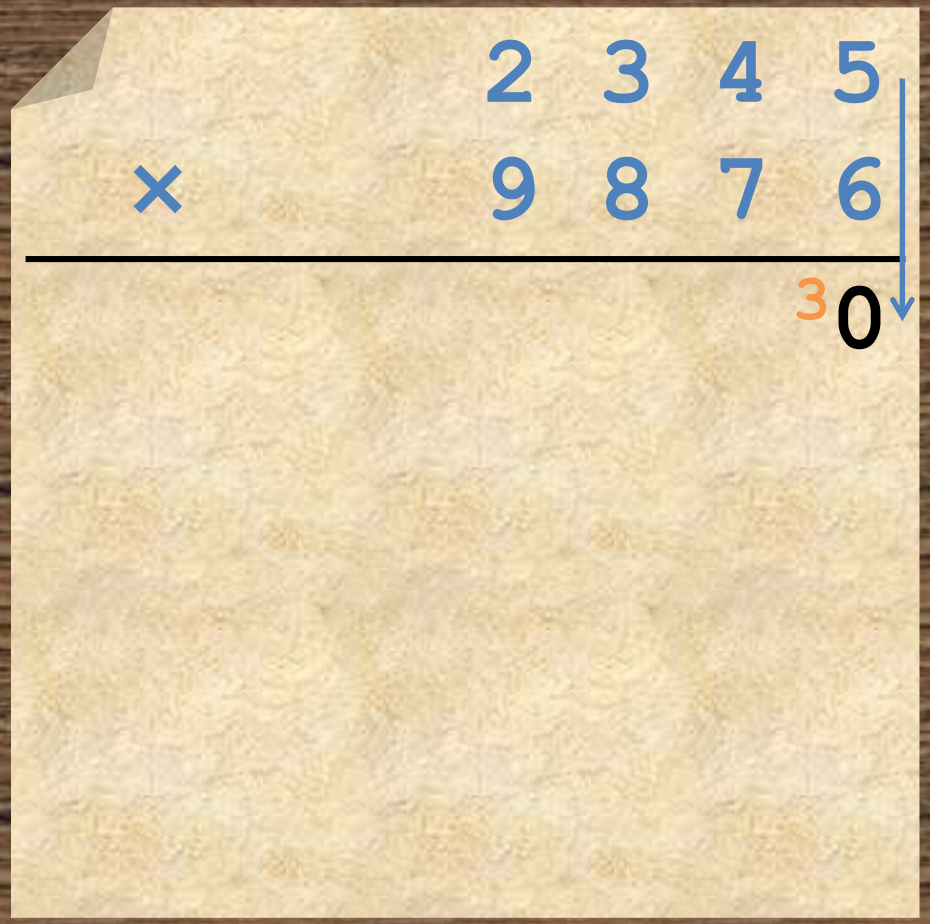
乘法的运算过程

北京大学·慕课
计算机组成
制作人：陆俊林

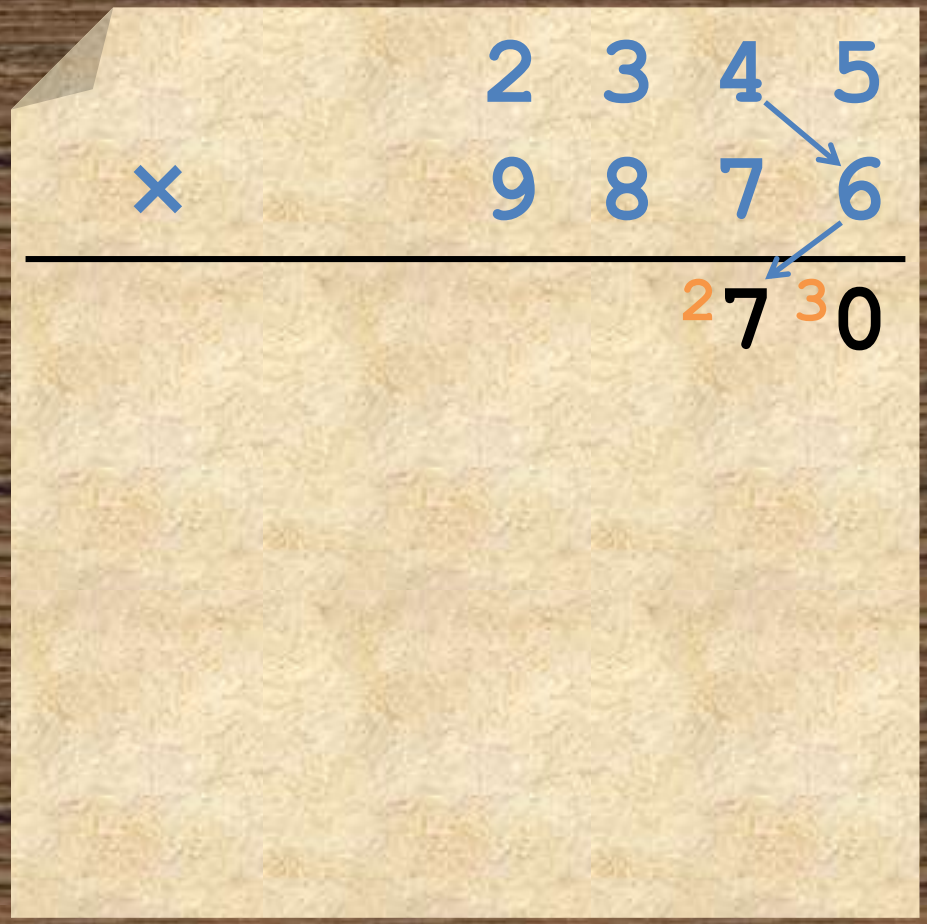
手工进行乘法运算

$$
\begin{array}{r}
2345 \\
\times\ 9876 \\
\hline
\end{array}
$$

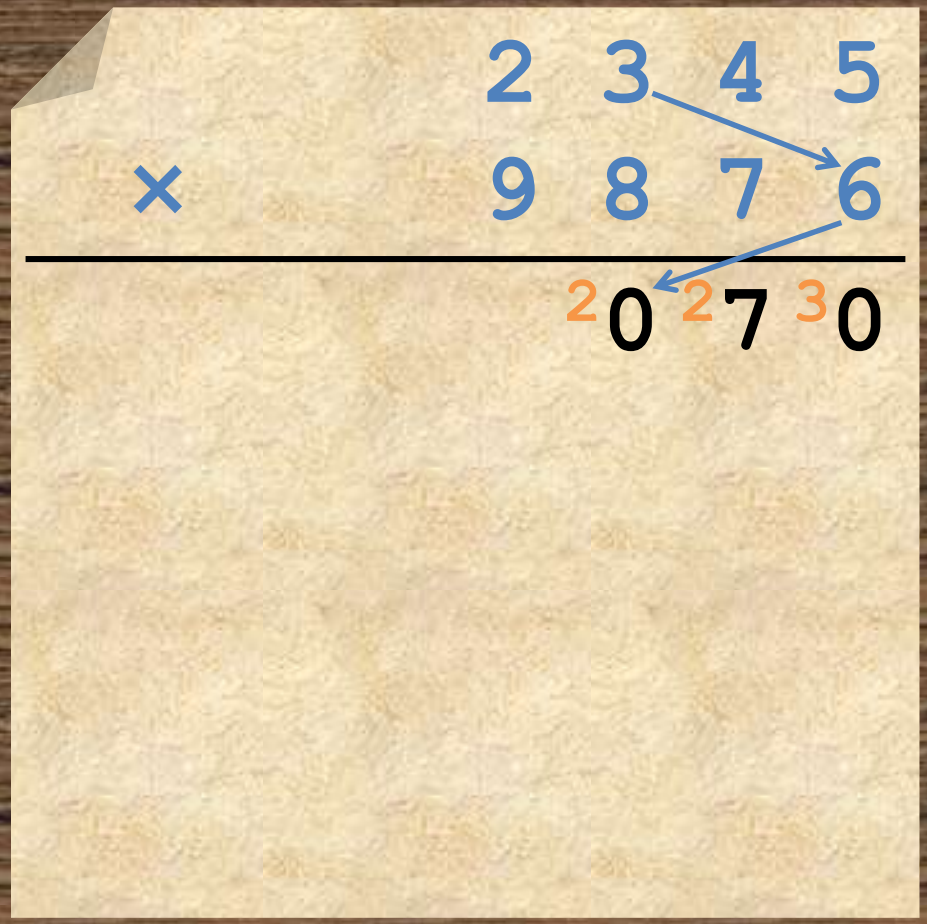
手工进行乘法运算



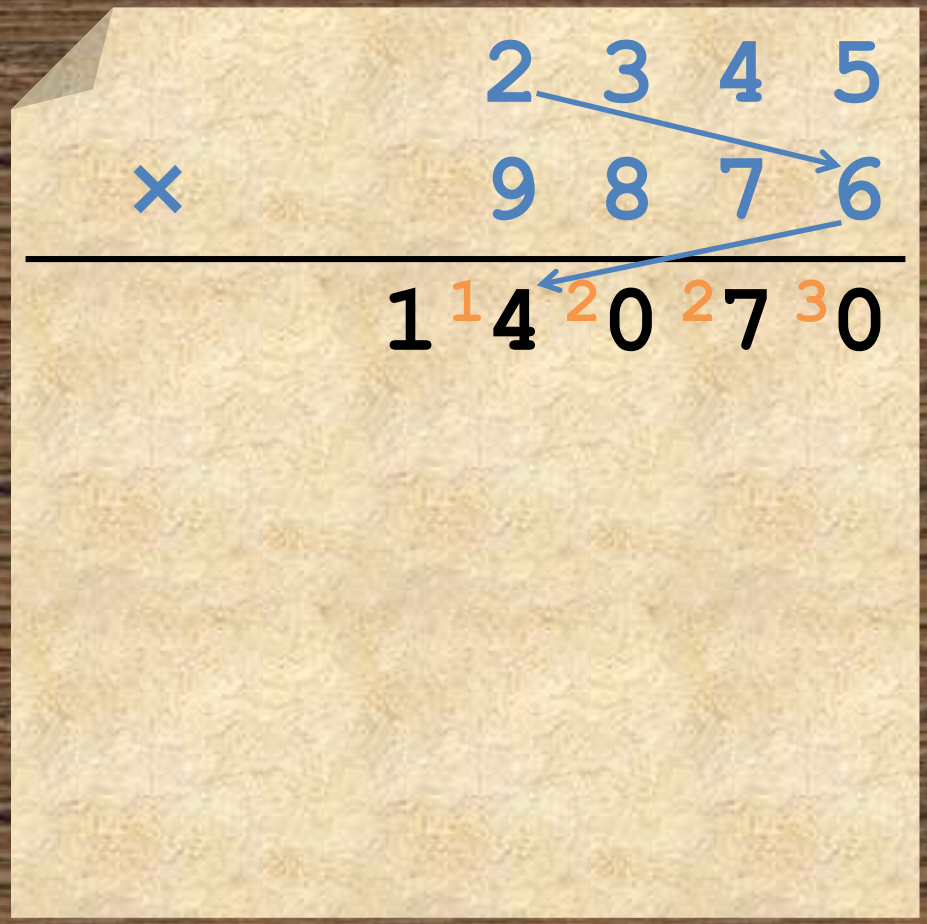
手工进行乘法运算



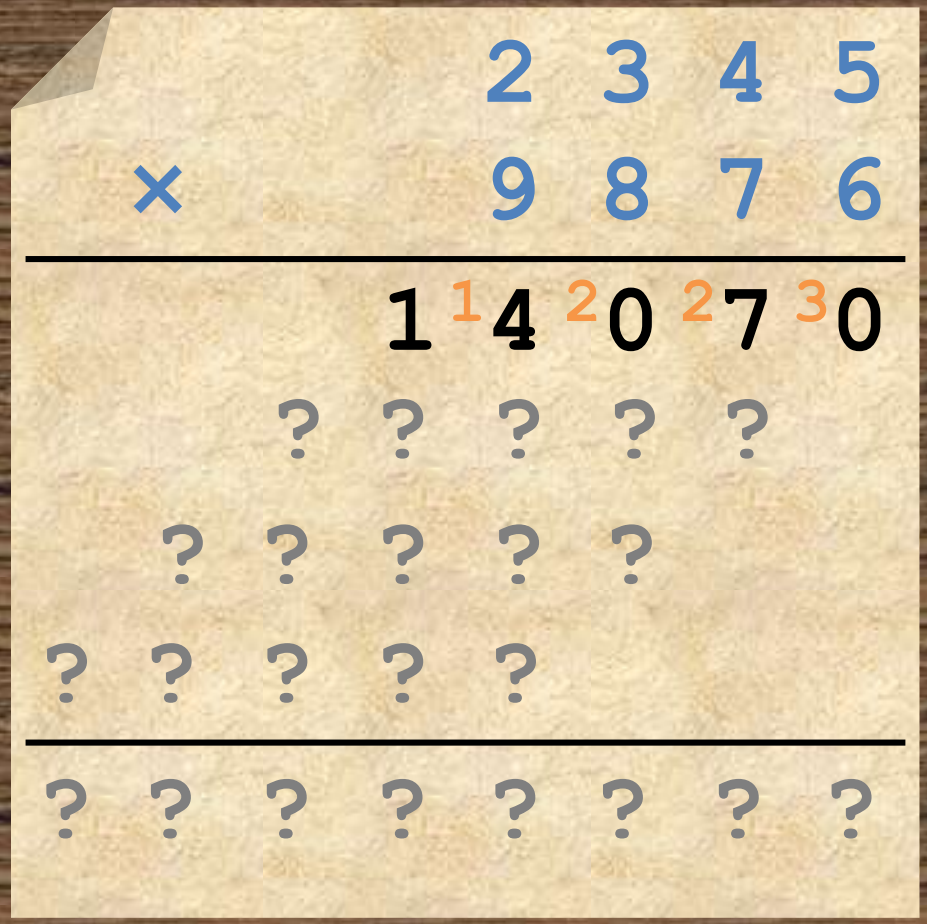
手工进行乘法运算



手工进行乘法运算

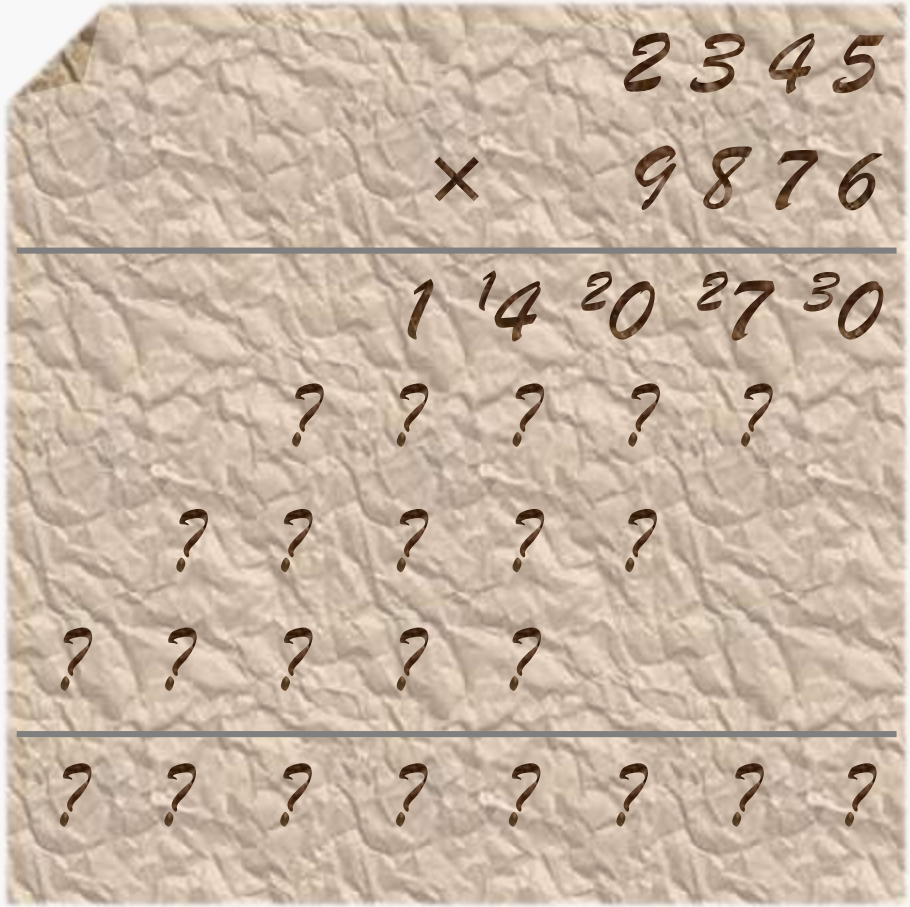


手工进行乘法运算



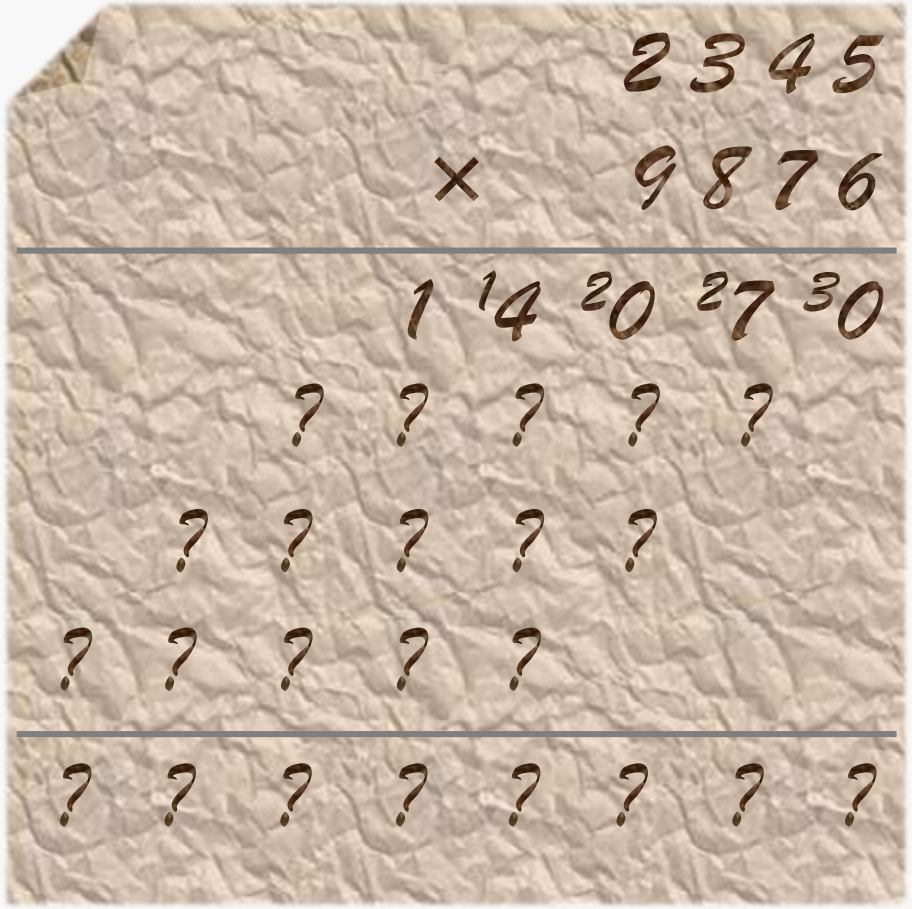
较为简单的数字

```
    1 0 0 0
×   1 0 0 1
-----------
    1 0 0 0
```



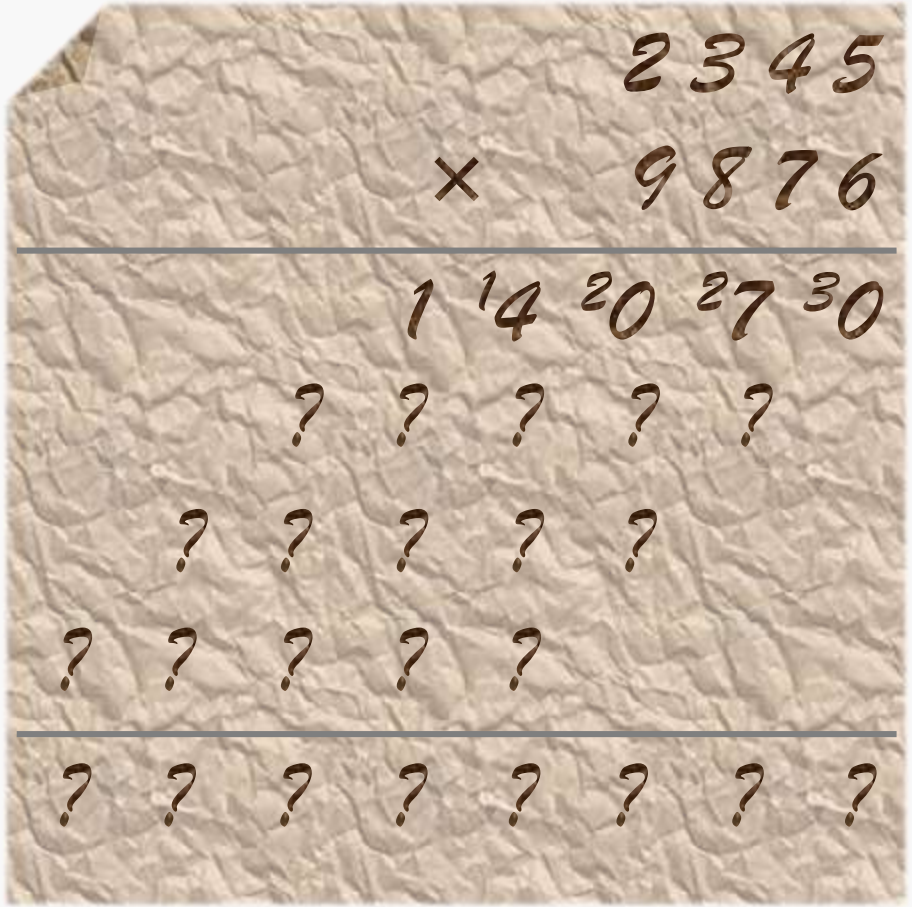
较为简单的数字

```
    1 0 0 0
×   1 0 0 1
-----------
    1 0 0 0
  0 0 0 0
```



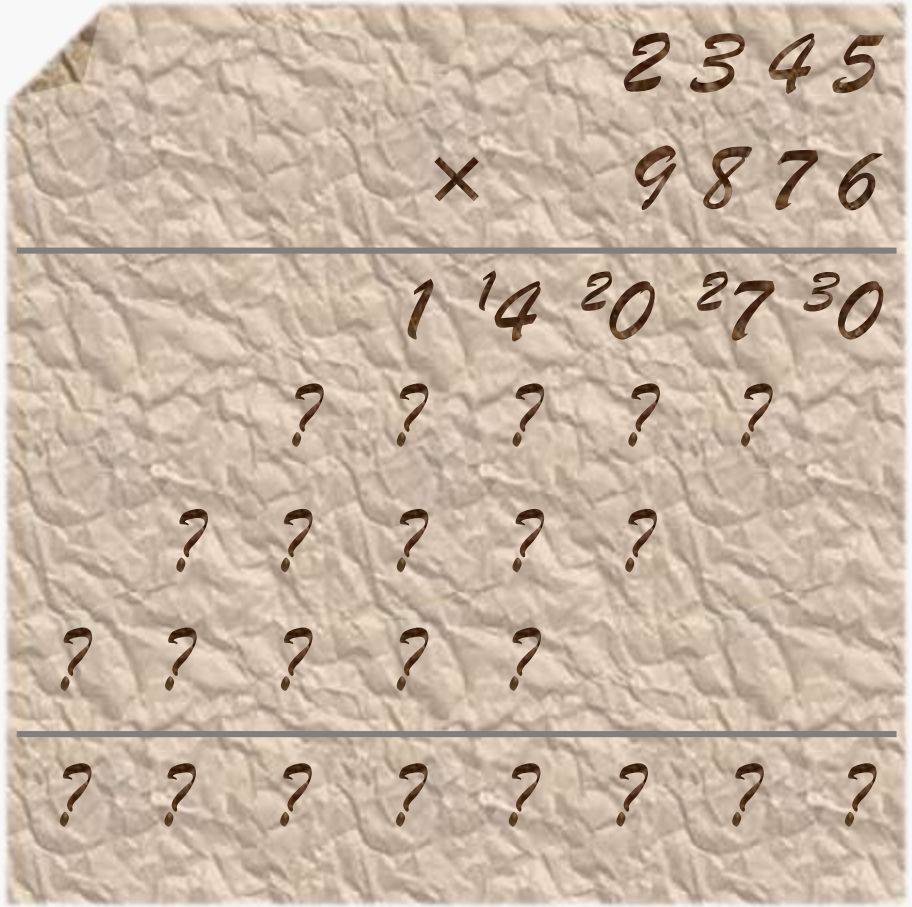
较为简单的数字

```
      1 0 0 0
  ×   1 0 0 1
-------------
      1 0 0 0
    0 0 0 0
  0 0 0 0
```



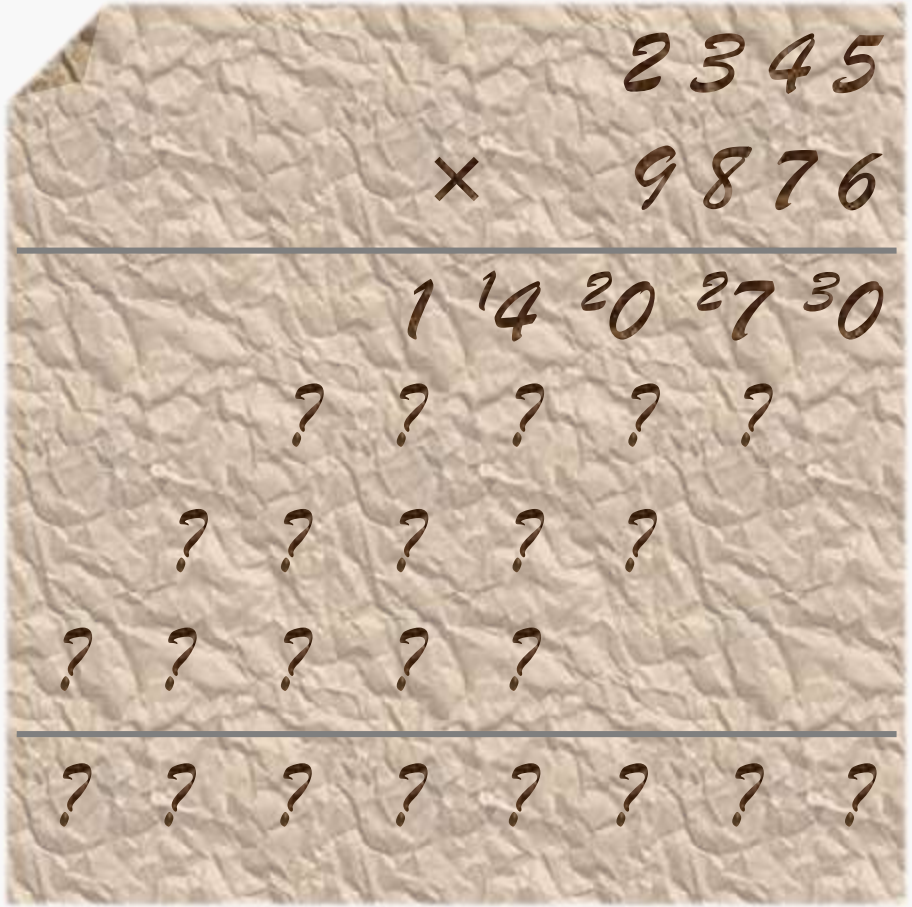
较为简单的数字

```
        1 0 0 0
  ×     1 0 0 1
-------------------
        1 0 0 0
      0 0 0 0
    0 0 0 0
  1 0 0 0
```



较为简单的数字

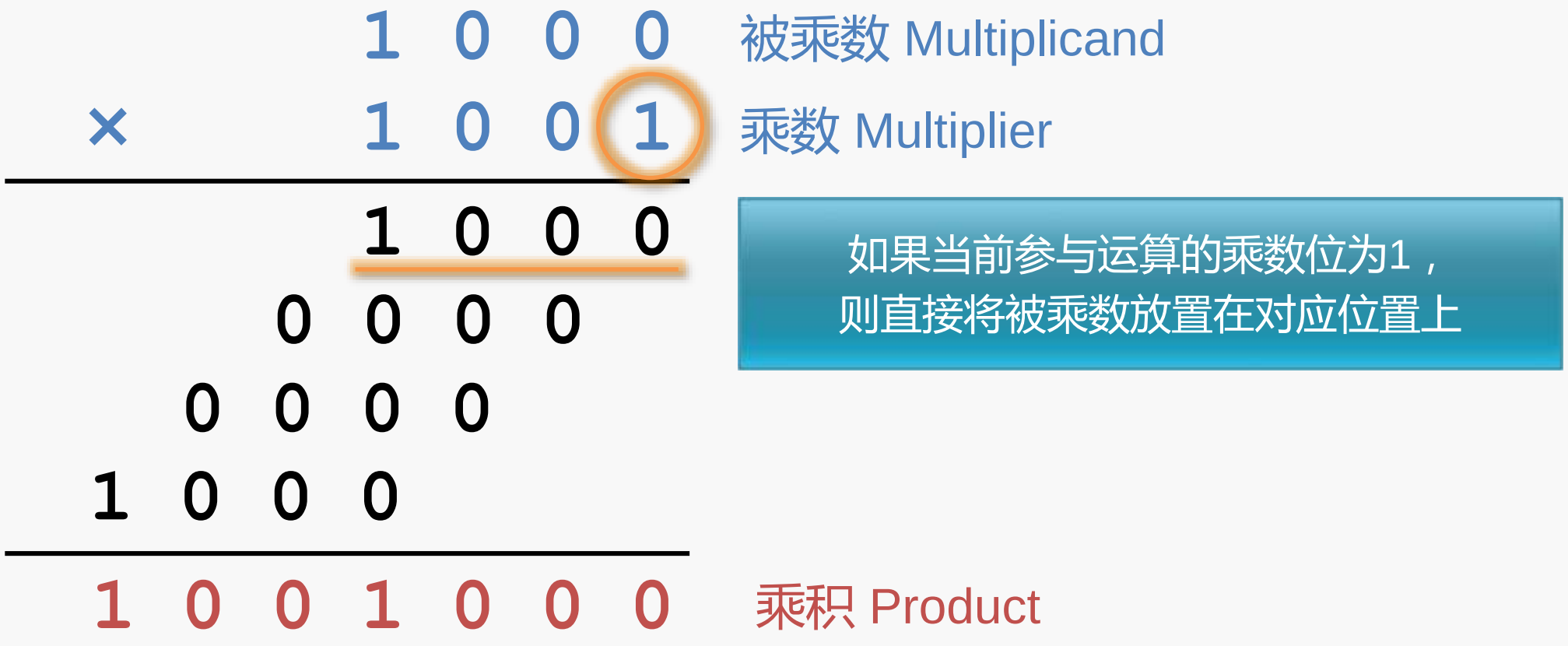
```
        1 0 0 0
  ×     1 0 0 1
-------------------
        1 0 0 0
      0 0 0 0
    0 0 0 0
  1 0 0 0
-------------------
  1 0 0 1 0 0 0
```



简化后的运算过程

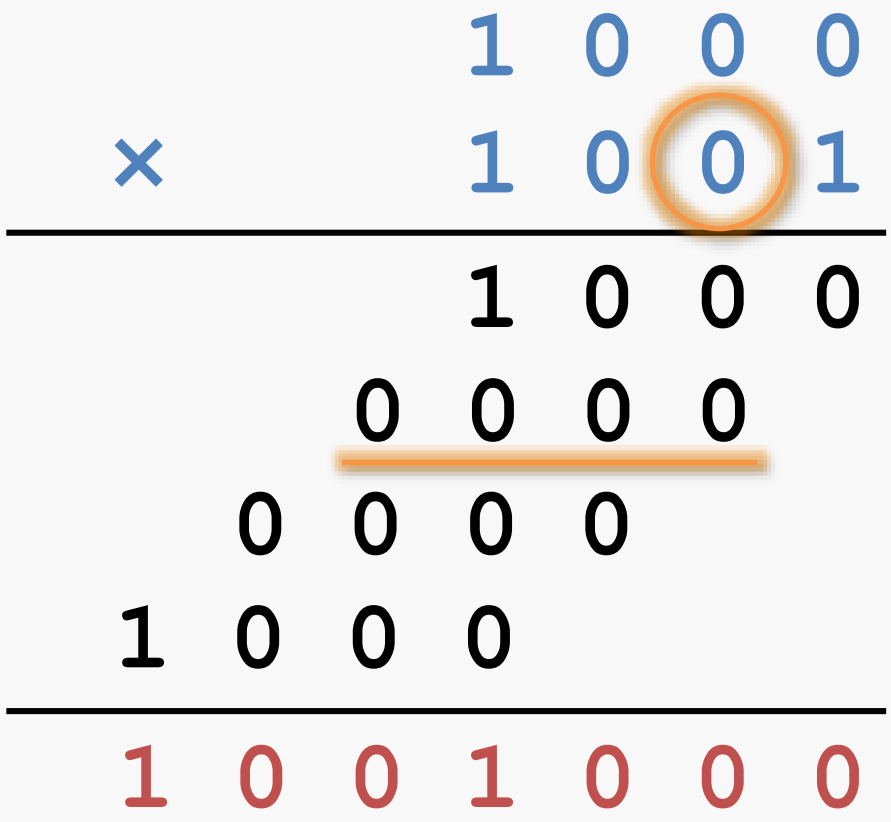
				1	0	0	0	被乘数 Multiplicand
×				1	0	0	1	乘数 Multiplier
				1	0	0	0	
			0	0	0	0		
		0	0	0	0			
1	0	0	0					
1	0	0	1	0	0	0		乘积 Product

简化后的运算过程



如果当前参与运算的乘数位为1，
则直接将被乘数放置在对应位置上

简化后的运算过程



被乘数 Multiplicand

乘数 Multiplier

如果当前参与运算的乘数位为1，
则直接将被乘数放置在对应位置上

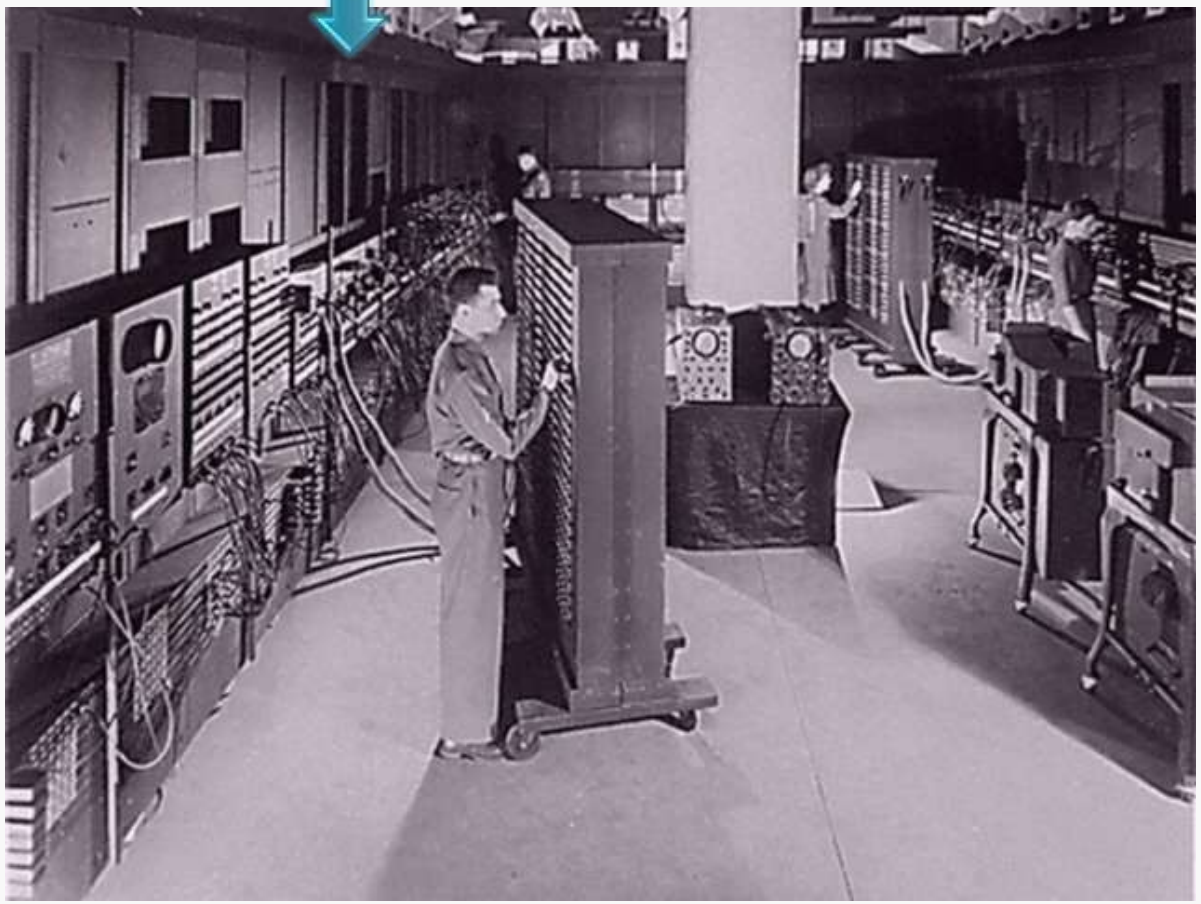
如果当前参与运算的乘数位为0，
则直接将“0”放置在对应位置上

乘积 Product

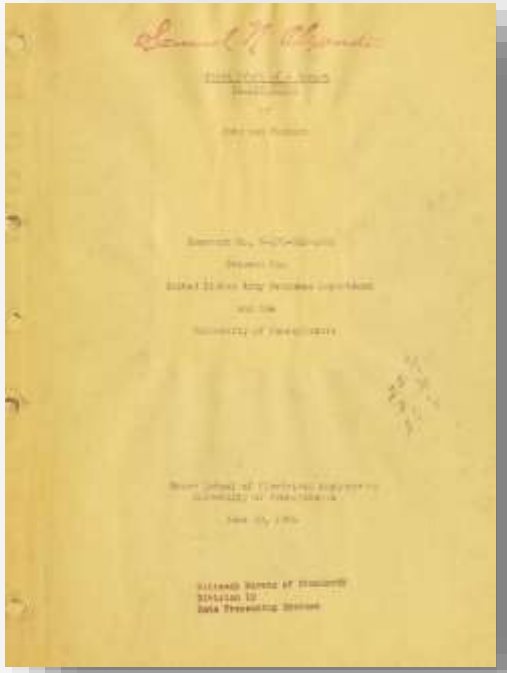
十进制和二进制运算的选择

采用十进制的ENIAC

采用二进制的EDVAC



十进制和二进制运算的选择

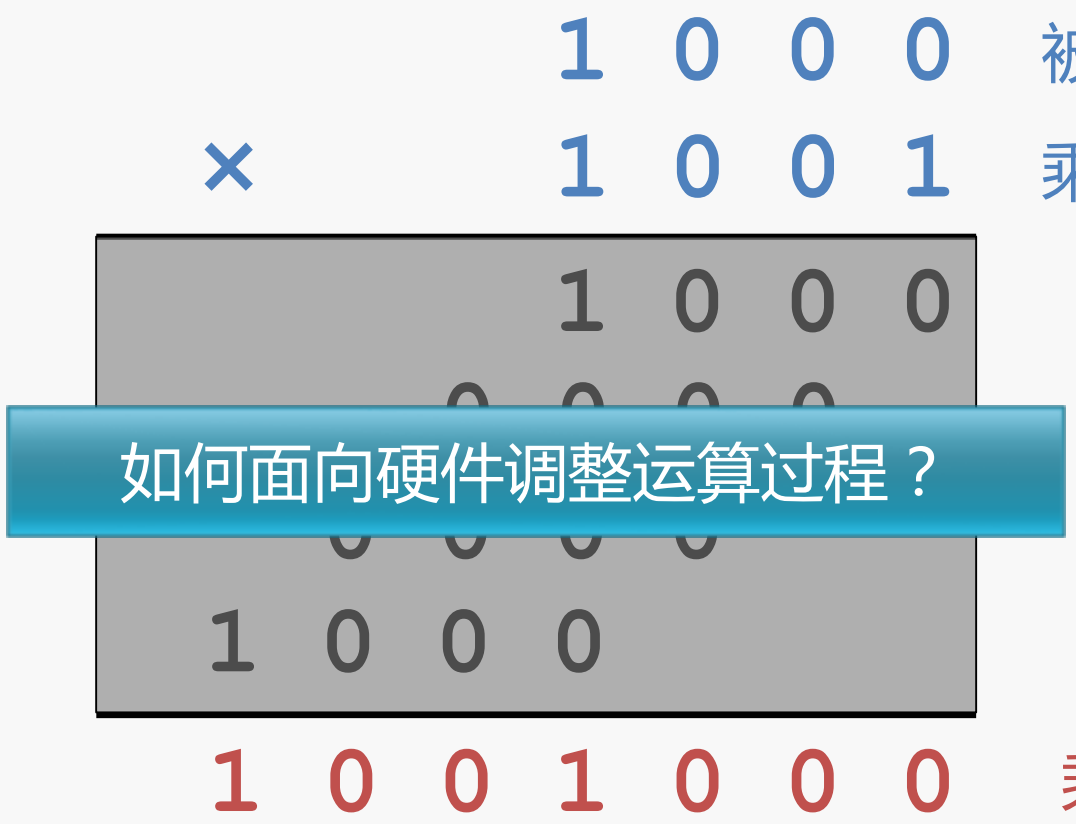


关于EDVAC的
报告草案
1945

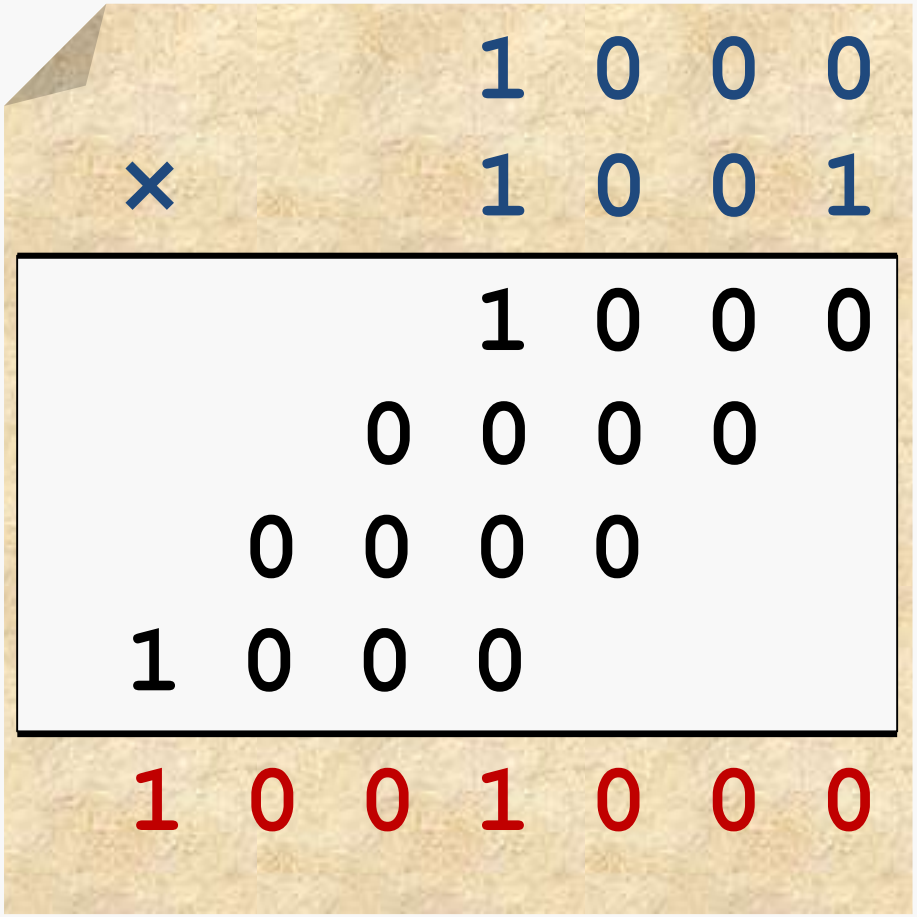
- 电子管是一种“全或无”设备（all-or-none），适合表示只有两个数值的系统，即二进制。
- 二进制可以大幅度地简化乘法和除法的运算过程。尤其是对于乘法，不再需要十进制乘法表，也不再需要两轮的加法。
- **必须要记住**，十进制才是适合人使用的。因此，输入输出设备需要承担二进制和十进制之间的转换工作。

约翰·冯·诺依曼
John Von Neumann
1903~1957

二进制乘法的运算过程


1 0 0 0
× 1 0 0 1
1 0 0 0
0 0 0 0
0 0 0 0
1 0 0 0
1 0 0 1 0 0 0
被乘数 Multiplicand
乘数 Multiplier
乘积 Product
如何面向硬件调整运算过程？

运算过程的进一步调整

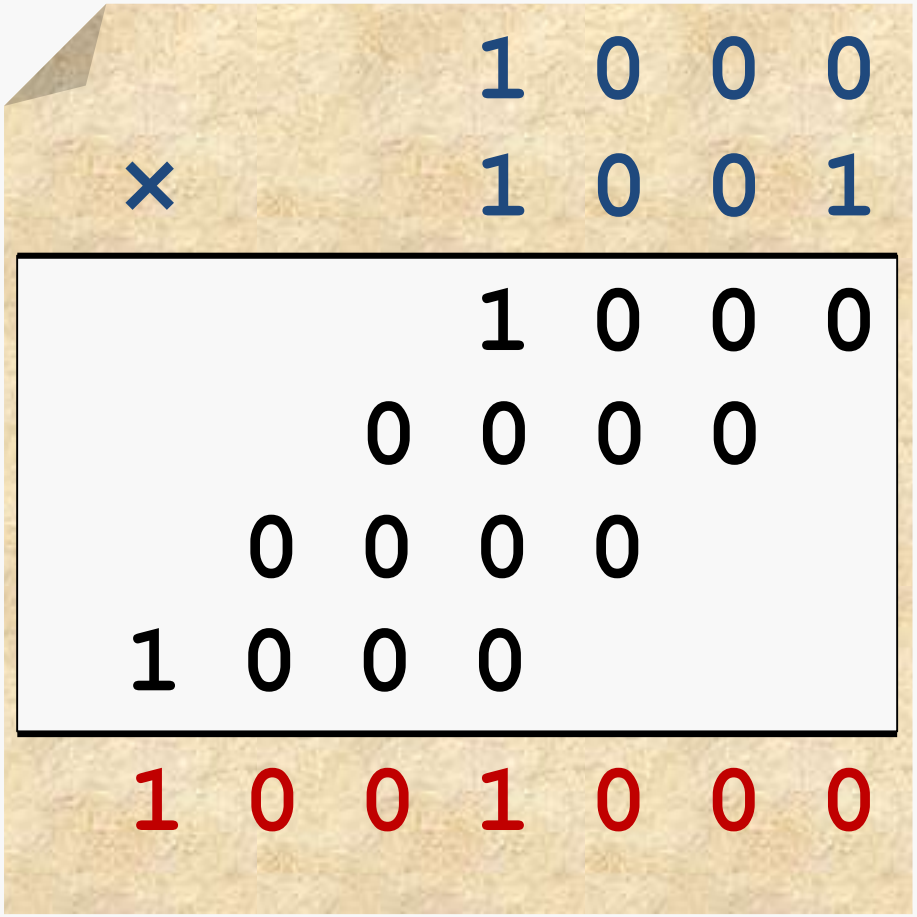


被乘数 Multiplicand

乘数 Multiplier

乘积 Product

运算过程的进一步调整



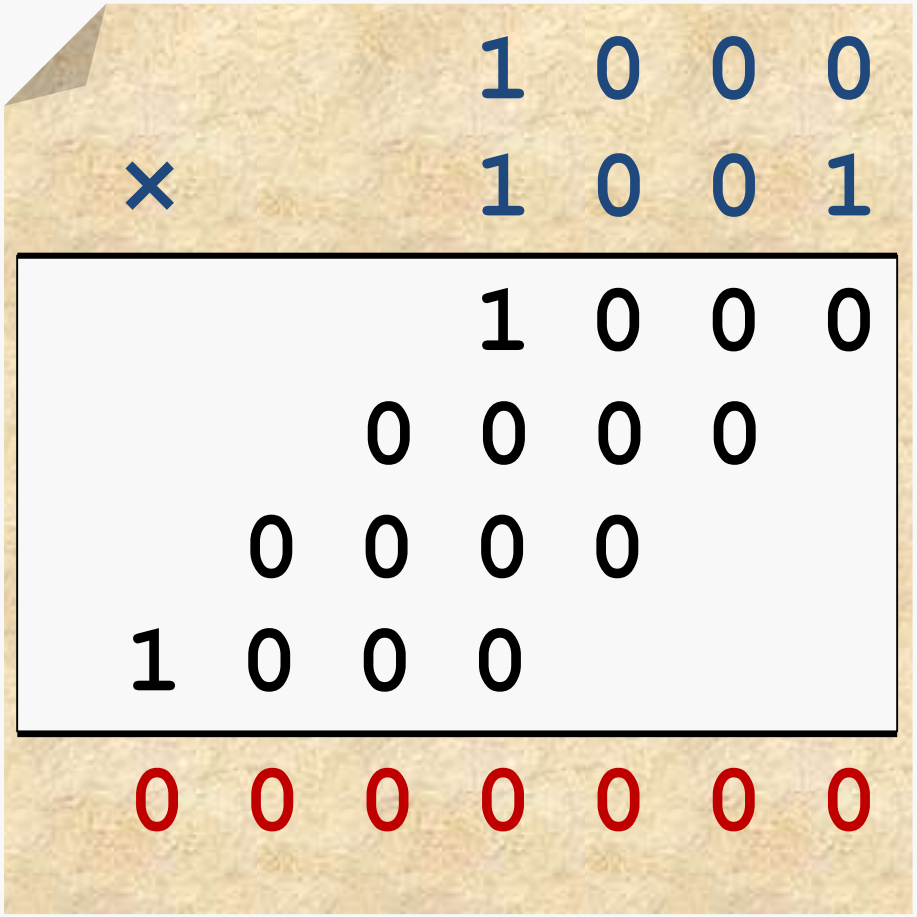
被乘数 Multiplicand

乘数 Multiplier

运算开始时，乘积记为“0”

乘积 Product

运算过程的进一步调整



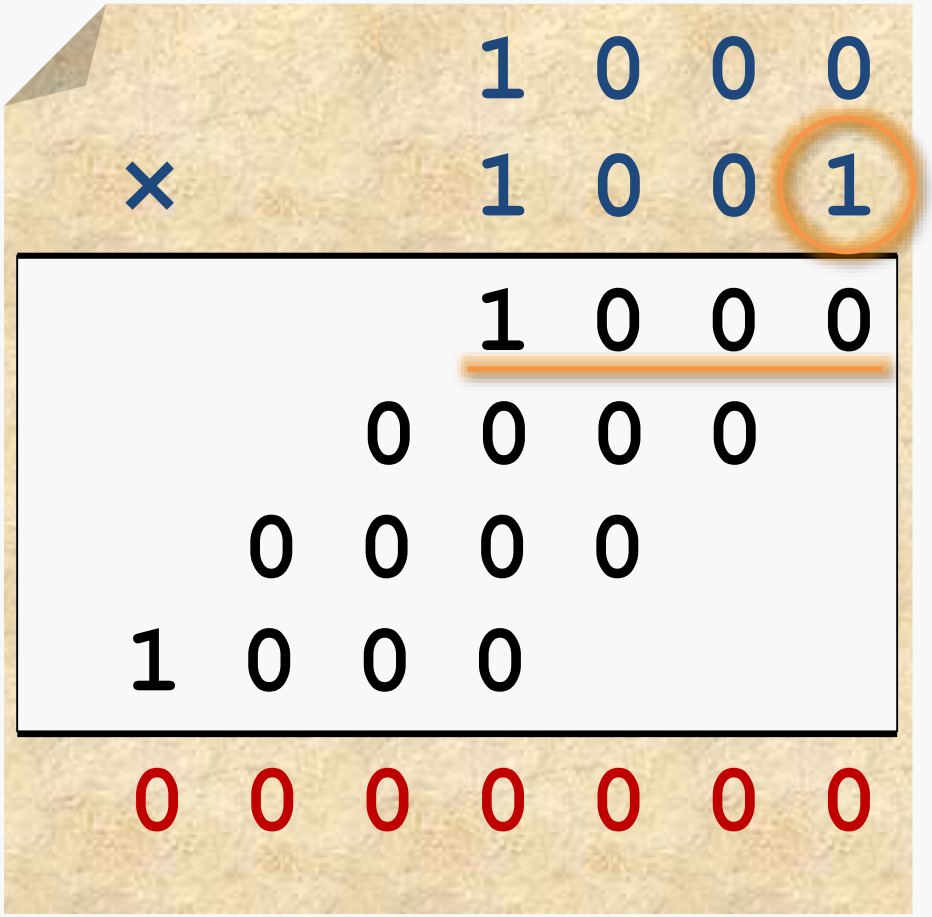
被乘数 Multiplicand

乘数 Multiplier

运算开始时，乘积记为“0”

乘积 Product

运算过程的进一步调整



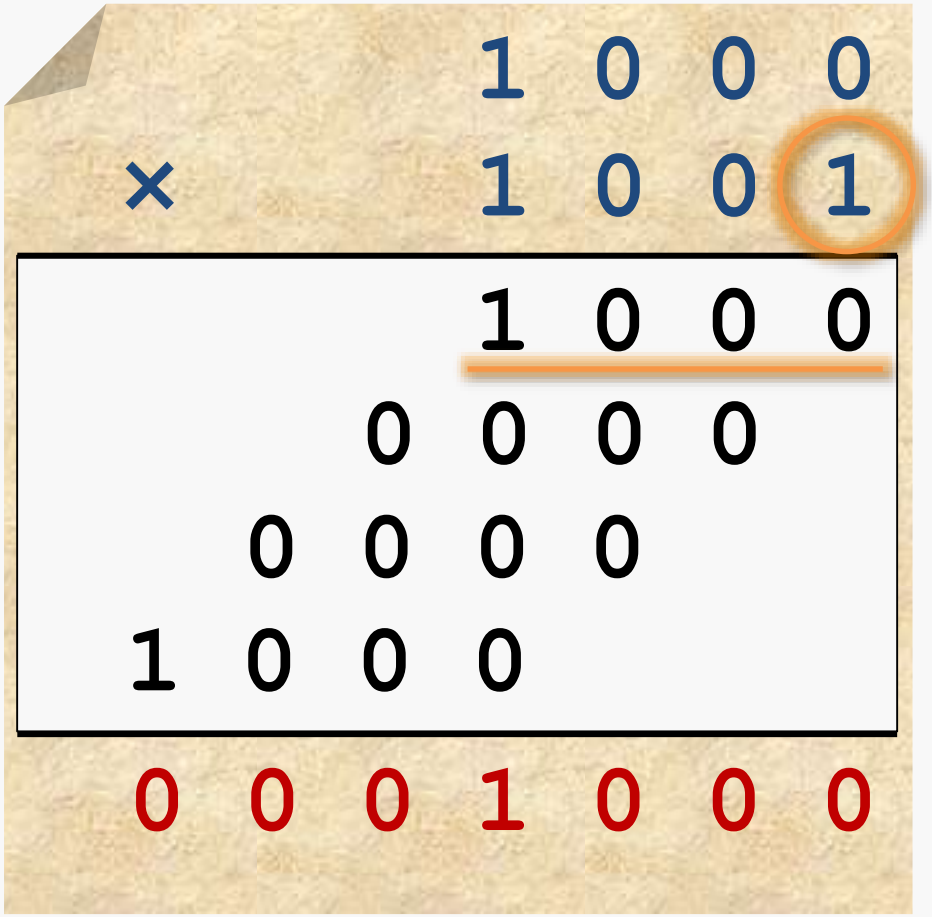
被乘数 Multiplicand

乘数 Multiplier

每个中间结果产生后
直接与当前的乘积累加

乘积 Product

运算过程的进一步调整



被乘数 Multiplicand

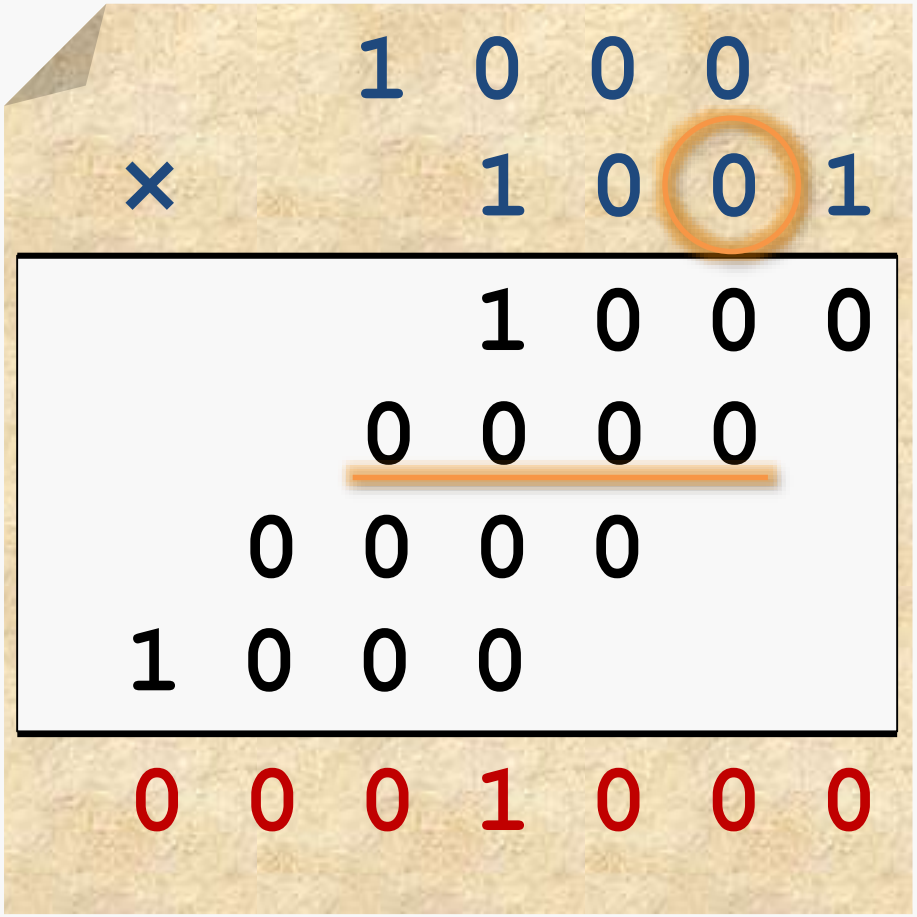
乘数 Multiplier

每个中间结果产生后
直接与当前的乘积累加

每产生一个中间结果
被乘数向左移动一位

乘积 Product

运算过程的进一步调整



被乘数 Multiplicand

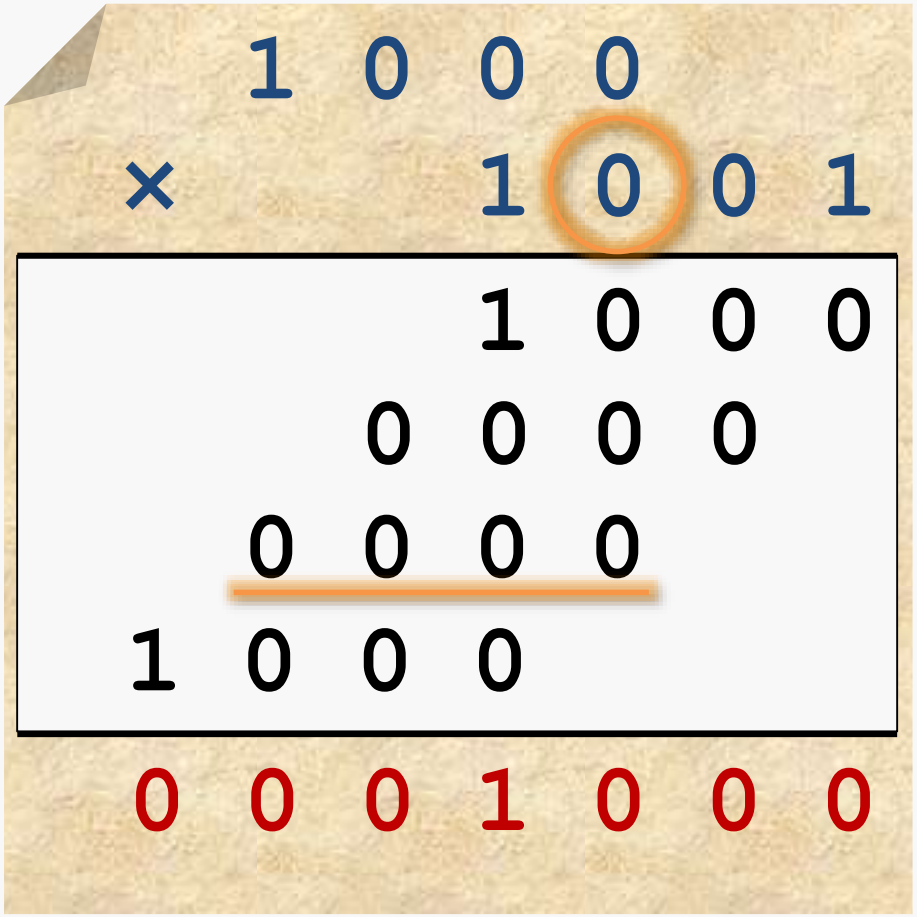
乘数 Multiplier

每个中间结果产生后
直接与当前的乘积累加

每产生一个中间结果
被乘数向左移动一位

乘积 Product

运算过程的进一步调整



被乘数 Multiplicand

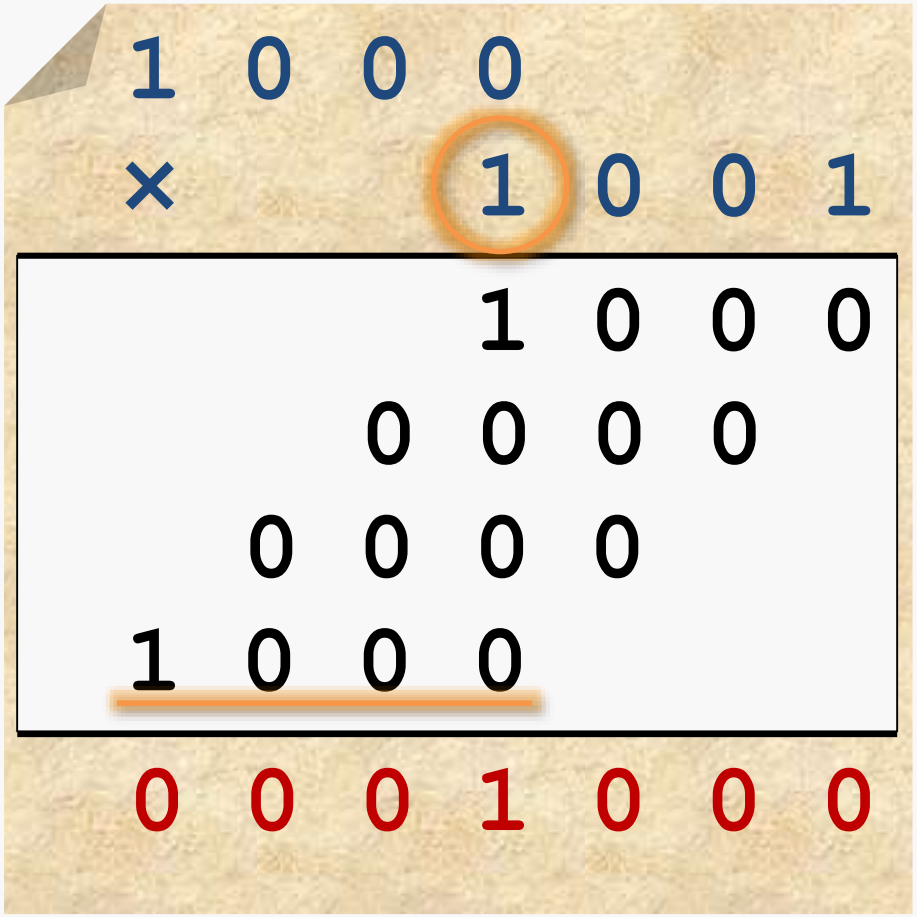
乘数 Multiplier

每个中间结果产生后
直接与当前的乘积累加

每产生一个中间结果
被乘数向左移动一位

乘积 Product

运算过程的进一步调整



被乘数 Multiplicand

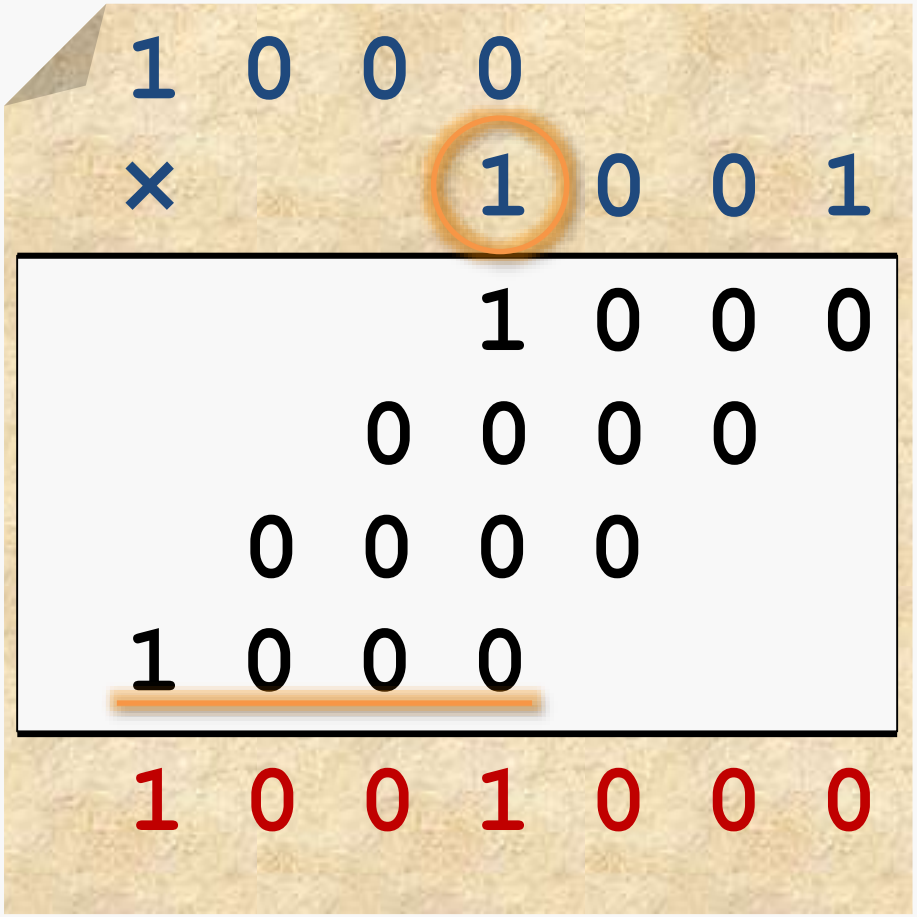
乘数 Multiplier

每个中间结果产生后
直接与当前的乘积累加

每产生一个中间结果
被乘数向左移动一位

乘积 Product

运算过程的进一步调整



被乘数 Multiplicand

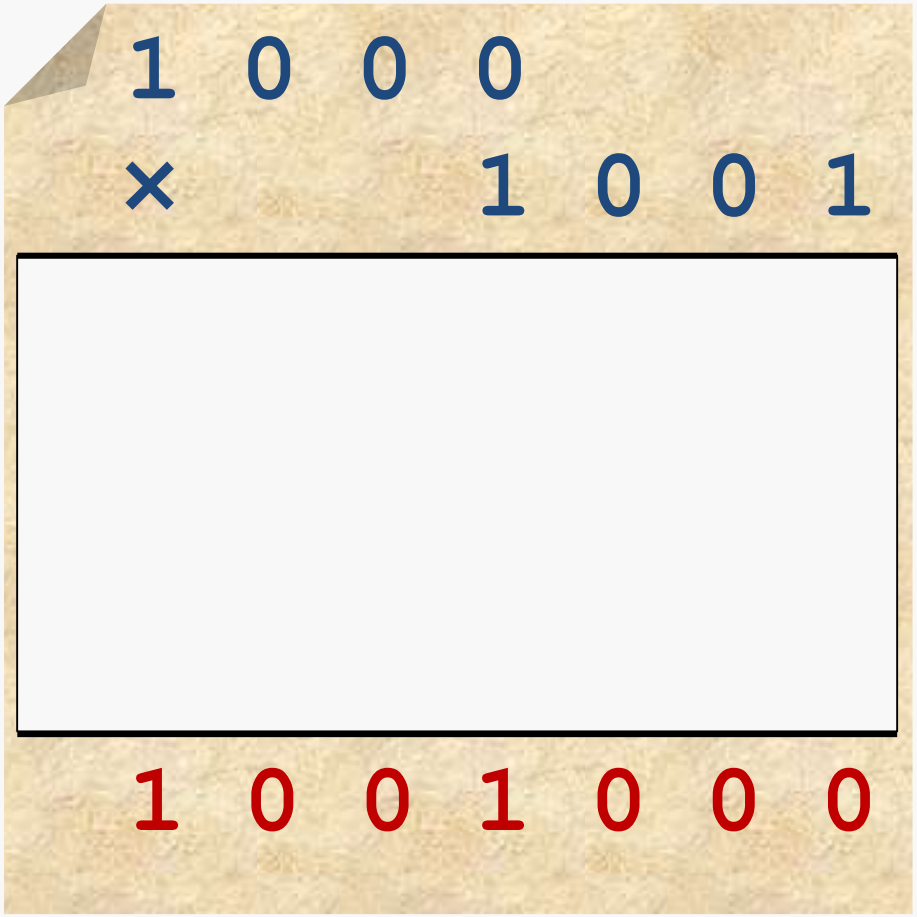
乘数 Multiplier

每个中间结果产生后
直接与当前的乘积累加

每产生一个中间结果
被乘数向左移动一位

乘积 Product

运算过程的进一步调整



被乘数 Multiplicand

乘数 Multiplier

适合硬件实现的运算过程！

乘积 Product

本节小结

乘法的运算过程

北京大学·慕课
计算机组成
制作人：陆俊林