Integer Multiplication

Bakh Khoussainov

Integer Multiplication problem.

Example:

$$
\begin{array}{cccccccc}
 & & & & 1 & 1 & 0 & 0 \\
\times & & & & 1 & 1 & 0 & 1 \\
\hline
 & & & & 1 & 1 & 0 & 0 \\
+ & & & 0 & 0 & 0 & 0 & \\
 & & 1 & 1 & 0 & 0 & & \\
 & 1 & 1 & 0 & 0 & & & \\
\hline
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0
\end{array}
$$

So, we added four "partial products".

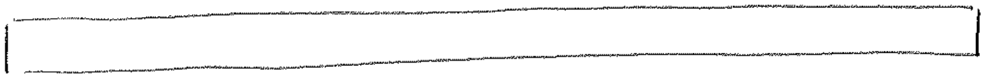
Question: Is there a cleverer way to multiply integers?

Let us assume we are in base 2.

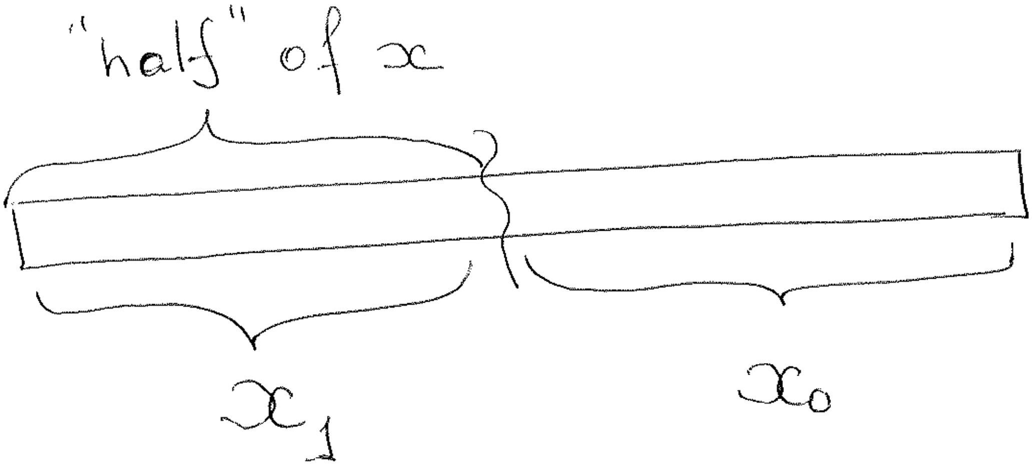
We are given two integers x and y.

Want to compute $x \cdot y$.

Take x:

x: 

"Half" this x:

x: 

So $x = x_1 \cdot 2^{n/2} + x_0$

where n is the number of bits in x.

For instance

$$x = 10111010101101 =$$

$$= \overbrace{1011101}^{x_1}0000000 +$$

$$0000000\underbrace{0101101}_{x_0}$$

We write y similarly

$$y = y_1 \cdot 2^{n/2} + y_0 .$$

So

$$x \cdot y = (x_1 \cdot 2^{n/2} + x_0)(y_1 \cdot 2^{n/2} + y_0) =$$

$$= x_1 \cdot y_1 \cdot 2^n + (x_1 y_0 + x_0 y_1) \cdot 2^{n/2} + x_0 y_0 .$$

So, we reduced one multiplication to four multiplications of $n/2$ bits of integers.

Thus, to compute $x \cdot y$ we have 4 recursive calls for computing:

(1) $x_1 y_1$

(2) $x_1 y_0$

(3) $x_0 y_1$

(4) $x_0 y_0$

Once, these are computed we can compute $x \cdot y$.

Note the following:

(1) $x_1 + x_0$, $y_1 + y_0$ both have $\frac{n}{2} + 1$ bits at most.

(2) The term

$$x_1 y_0 + x_0 y_1$$

is present in direct computation of $x \cdot y$.

Thus, to compute $x \cdot y$ we can proceed as follows:

(1) Compute $x_1 y_1$

(2) Compute $x_0 y_0$

(3) Compute $(x_0+x_1)(y_0+y_1)$

So we can calculate $x_1 y_0 + x_0 y_1$ as

$$(x_1+x_0)(y_1+y_0) - x_1 y_1 - x_0 y_0.$$

Thus, we have 3 recursive calls to multiply n-bit integers x and y.

Each of these 3 recursive calls compute the product of $n/2$ bit of integers.

Those are then put together using + AND −, to compute $x \cdot y$.

Recursive-Multiply (x,y) algorithm:

Write $x = x_1 2^{n/2} + x_0$,

$y = y_1 \cdot 2^{n/2} + y_0$.

Compute $x_0 + x_1$, $y_0 + y_1$.

$p =$ Recursive-Multiply $(x_0 + x_1, y_0 + y_1)$

$x_1 y_1 =$ Recursive-Multiply (x_1, y_1)

$x_0 y_0 =$ Recursive-Multiply (x_0, y_0)

Return

$x_1 y_1 \cdot 2^n + (p - x_1 y_1 - x_0 y_0) + x_0 y_0$