than the bit operations; in general, it is the latter which takes by far the most time.

Next, let's examine the process of *multiplying* a k-bit integer by an ℓ-bit integer in binary. For example,

$$\begin{array}{r} 11101 \\ \underline{\quad 1101} \\ 11101 \\ 111010 \\ \underline{11101\quad\ } \\ 101111001 \end{array}$$

Suppose we use this familiar procedure to multiply a k-bit integer n by an ℓ-bit integer m. We obtain at most ℓ rows (one row fewer for each 0-bit in m), where each row consists of a copy of n shifted to the left a certain distance, i.e., with zeros put on at the end. Suppose there are $\ell' \leq \ell$ rows. Because we want to break down all our computations into bit operations, we cannot simultaneously add together all of the rows. Rather, we move down from the 2nd row to the ℓ'-th row, adding each new row to the partial sum of all of the earlier rows. At each stage, we note how many places to the left the number n has been shifted to form the new row. We copy down the right-most bits of the partial sum, and then add to n the integer formed from the rest of the partial sum — as explained above, this takes k bit operations. In the above example 11101×1101, after adding the first two rows and obtaining 10010001, we copy down the last three bits 001 and add the rest (i.e., 10010) to $n = 11101$. We finally take this sum $10010 + 11101 = 101111$ and append 001 to obtain 101111001, the sum of the $\ell' = 3$ rows.

This description shows that the multiplication task can be broken down into $\ell' - 1$ additions, each taking k bit operations. Since $\ell' - 1 < \ell' \leq \ell$, this gives us the simple bound

$$\text{Time(multiply integer } k \text{ bits long by integer } \ell \text{ bits long)} < k\ell.$$

We should make several observations about this derivation of an estimate for the number of bit operations needed to perform a binary multiplication. In the first place, as mentioned before, we counted only the number of bit operations. We neglected to include the time it takes to shift the bits in n a few places to the left, or the time it takes to copy down the right-most digits of the partial sum corresponding to the places through which n has been shifted to the left in the new row. In practice, the shifting and copying operations are fast in comparison with the large number of bit operations, so we can safely ignore them. In other words, we shall *define* a "time estimate" for an arithmetic task to be an upper bound for the number of bit operations, without including any consideration of shift operations,