Hamming Weight via Word Packing.

Consider our input string $x$ of length $k$ and let $z = \lfloor 2 + \lg k \rfloor$. We assume that $k + 1$ is not a power of 2 (otherwise we shorten $x$ by 1 by computing $x \% 2$ and one shift right) and that $z$ and $k$ are relatively prime (otherwise set $z = z - 1$).

The key idea is to transform $x$ into a bit string that still fits in a machine word and has the same number of ones as $x$ but all the ones are in locations that are multiples of $z$. Then all we need to do is use the modulo operator ($\%(2^z - 1)$) on the new string (in $O(1)$ time).

1. Set $A$ to be a string with $z$ ones and $k$ zeros between any two ones.

2. Compute $y = xA$ ($y$ is a bitstring that consists of $z$ copies of $x$).

3. Set $B$ to be a string with $k$ ones and $z$ zeros between any two ones.

4. Return $(y \text{ AND } B)\%(2^z - 1)$.

AND-ing $y$ with $B$ is equivalent to partitioning $y$ to consecutive blocks of length $z$ and replacing every block with a 000..01 iff the rightmost bit of the block is 1. Because $z$ and $k$ are relatively prime, this operation makes sure that every 1 bit in $x$ corresponds to some unique 1 bit in $(y \text{ AND } B)$ that appears in a location that are multiples of $z$. Thus, $(y \text{ AND } B)\%(2^z - 1)$ gives the answer.