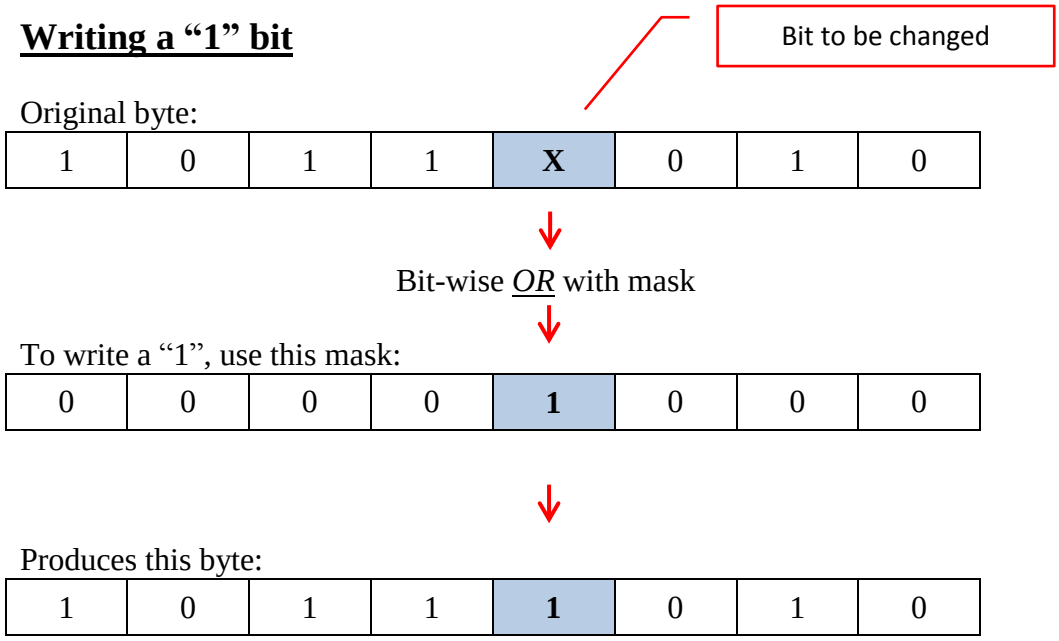
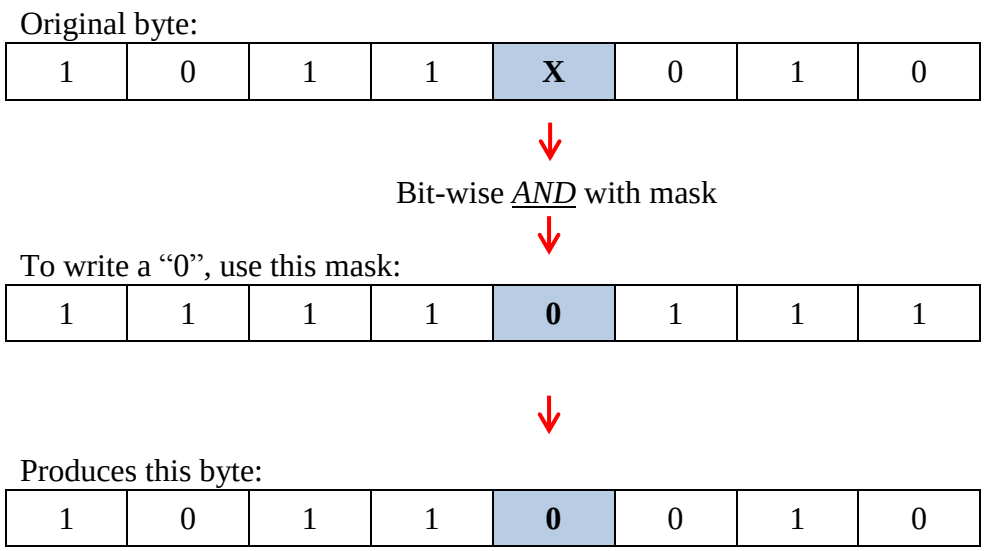
Using Masks to Manipulate Bits in a Byte
(Just the Basics)

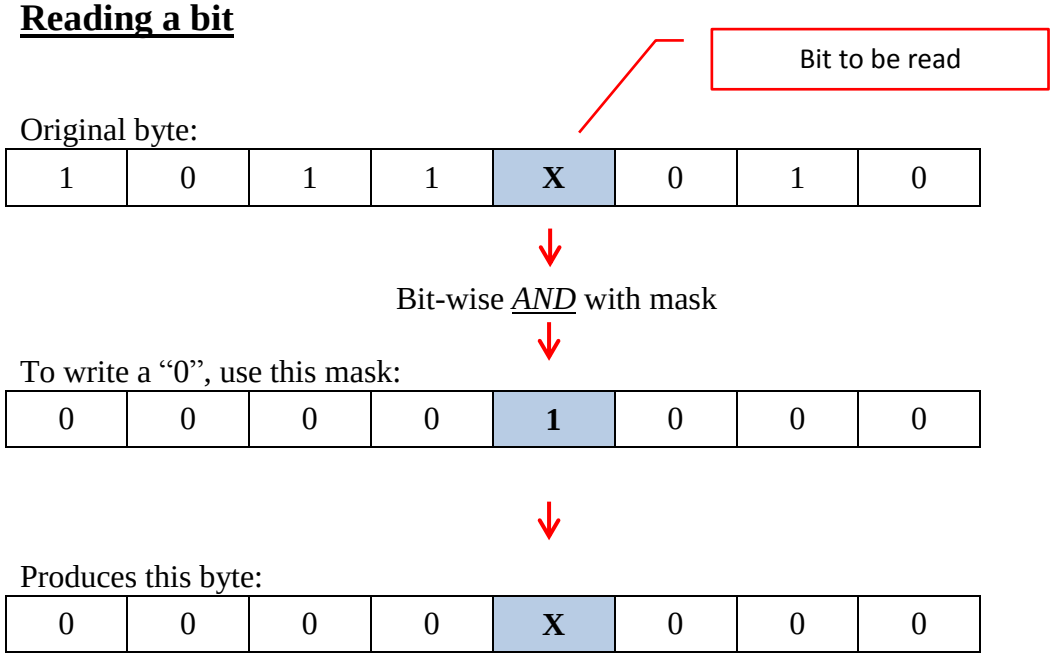
Writing a "1" bit



Writing a "0" bit



Reading a bit



If the resulting byte is non-zero, the bit was a "1"

Working With Bit Masks

There are many variations on bit masking:

- These examples use bytes as illustration, but bit masks are usually used with *int* variables.
- Bit masks can be saved in arrays, making it easier to manipulate bit states in loops.
- In the "write" example above the bit-wise NOT operator (~) can be applied to the "1" mask to produce the "0" mask.
- A bit-wise XOR mask can be used to "flip" a bit state.
- Bit shift operators can be used with a single mask in a loop to move an individual flag from bit to bit.
- Multiple bits can be set in the same operation by combining masks.