First and foremost, this book is NOT about hacking in the modern sense of finding out weaknesses in a computer system and exploiting them. In the early days of computers, a hacker was someone who delighted in subtle programming tricks and small algorithms that could be used to make their computer code “tighter” and more efficient.

Hacker's Delight really is a great book. Just to give you a feel for it, it’s the first book of this type that I’ve ever seen on Amazon in which every reviewer (and there are a bunch of them) has awarded five stars.

On the one hand, the tricks in this book are becoming less relevant as computational resources – in the form of memory and performance – increase. Many of today’s programmers work at a high level of abstraction and are conceptually a long way from the “machine”. Also, in many cases, a slight inefficiency in the code is of less importance than getting the application out of the door in a timely manner.

On the other hand, if you are working on embedded systems with limited performance and memory resources, and if you are working “close to the metal”, then the tips and tricks in this book are for you.

Most of the time the author assumes a 32-bit machine and describes his tips and tricks in a generic computer algebra; more complex tricks are presented in C; but all of these techniques could be easily adapted to machines of different word lengths (8-bits, 16-bits, 64-bits, etc.) and/or implemented in assembly language if required.

As a simple tempting taster, let’s consider the very first page of Chapter 2: Basics (Chapter 1 is the introduction where the author defines things like the notations to be used and the syntax of his computer algebra).

Note that in the following discussions:

- $-x$ = Arithmetic negation
- $\neg x$ = Bitwise NOT (ones complement)
- $x \& y$ = Bitwise AND
- $x | y$ = Bitwise OR
- $x ^ y$ = Bitwise Exclusive OR

OK, the first part of the first page reads as follows:
Manipulating Rightmost Bits

Use the following formula to turn off the rightmost 1-bit in a word, producing 0 if none (e.g., 01011000 => 01010000):

\[ x \& (x - 1) \]

This may be used to determine if an unsigned integer is a power of 2 or is a 0; apply the formula followed by a 0-test on the result.

Similarly, the following formula can be used to test if an unsigned integer is of the form \(2^n - 1\) (including all 0 or all 1’s):

\[ x \& (x + 1) \]

Use the following formula to isolate the rightmost 1-bit, producing 0 if none (e.g., 01011000 => 00001000):

\[ x \& (-x) \]

Use the following formula to isolate the rightmost 0-bit, producing 0 if none (e.g., 10100111 => 00001000):

\[ \sim x \& (x + 1) \]

Use one of the following formulas to form a mask that identifies the trailing 0’s, producing all 1’s if \(x = 0\) (e.g., 01011000 => 00000111):

\[ \sim x \& (x - 1), \text{ or } \sim(x \mid -x), \text{ or } (x \& -x) - 1 \]

The first formula has some instruction-level parallelism.

I haven’t finished yet. Remember that we are still on the very first page of the main body of the book. At the bottom of the page is one more formula to form a mask that identifies the rightmost 1-bit and the trailing 0’s, producing all 1’s if \(x = 0\) (e.g., 01011000 => 00001111). This formula is as simple as its companions shown above ... I’ll leave it as a little teaser for you to play with.

As one reviewer on Amazon said “If you approach optimization as a puzzle, wherein the solution is its own reward, this book is indeed a compendium of delights.”

I totally agree, and I highly recommend this book to anyone who is interested in knowing the "tricks of the trade." This is of especially interest to embedded software developers working with limited resources, but it is also of interest to any software developer who takes satisfaction in making his or her code as “tight” and efficient as possible.

If you found this article to be interest, visit [Microcontroller / MCU Designline](http://www.designline.com) where – in addition to my blogs on all sorts of "stuff" (also check out my [Max’s Cool Beans](http://www.maxcoolbeans.com) blog) – you will find the latest and greatest design, technology, product, and news articles with regard to all aspects of
designing and using microcontrollers.

Also, you can obtain a highlights update delivered directly to your inbox by signing up for my weekly newsletter – just Click Here to request this newsletter using the Manage Newsletters tab (if you aren't already a member you'll be asked to register, but it's free and painless so don't let that stop you [grin]).

Last but certainly not least, make sure you check out all of the discussions and other information resources at MicrocontrollerCentral.com, including blogs by yours truly.