

# Re: Integer arithmetic, multiplication overflow

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math/2005-06/msg01272.html>

---

- *From:* JEMebius <jemebius@xxxxxxxxxx>
  - *Date:* Wed, 08 Jun 2005 22:12:32 +0100
- 

If the result register of an integer multiplication instruction is twice as wide as the factor registers then overflow will never occur. I guess that in all processor architectures this is the case.

For instance, in IBM S/360 the integer multiplication instruction reads in assembler language format "M R1,D2(X2,B2)". Here the first operand R1 (an even number) represents an even-odd pair of 32-bit registers. The odd-numbered register R1+1 contains the first factor, a 32-bit signed integer. The second operand D2(X2,B2) represents the second factor, a 32-bit signed integer in main storage. The outcome of the multiplication is placed as a 63-bit signed integer in the two registers R1 and R1+1. One checks easily that overflow never occurs, neither at the  $2^{63}-1$  end, nor at the  $-2^{63}$  end.

BTW, You might get more and better response in a computer-oriented newsgroup.

Johan E. Mebius

Tomek wrote:

Hi,

I trying to solve the following thing:  
Having two finite precision unsigned integer numbers A and B,  
and knowing that maximal represented number is X,  
calculate  $A*B - X$ , in case of  $A*B$  overflow. Use only  
basic algebraic operations.

## Re: Integer arithmetic, multiplication overflow

Any hints welcome.  
Thanks,

Tomek