Subject: Re: counting bits
Posted by Craig Markwardt on Tue, 18 Feb 2003 04:45:42 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@as.arizona.edu> writes:
>
> For a 2048x2048 array of random long integers, this is 4-6 times as
> fast as a more traditional shift method, like the one David suggests.
> And here's an ultra-bizzarre one, which requires no look-up table, but
> holds its own against the LUT method (note the original array is
> destroyed):
>
> arr = ishft(arr AND 'AAAAAAAA'XUL,-1) + (arr AND '55555555'XUL)
> arr = ishft(arr AND 'CCCCCCCC'XUL,-2) + (arr AND '33333333'XUL)
> arr = ishft(arr AND 'F0F0F0F0'XUL,-4) + (arr AND '0F0F0F0F'XUL)
> arr = ishft(arr AND 'FF00FF00'XUL,-8) + (arr AND '00FF00FF'XUL)
> arr = ishft(arr AND 'FFFF0000'XUL,-16) + (arr AND '0000FFFF'XUL)
> tot=ulong(total(arr))
>
> See if you can figure that one out ;).

Very wicked!

Okay, here's a problem I've always solved by the brute force method:

 * what is the lowest / highest bit position set in an integer?

For example, a the lowest bit position in the binary number 11010100
is 2 (bit positions are labeled starting with 0 of course). The brute
force method involves testing each bit in succession using something
like (VALUE AND 2L^I) for each I, until a set bit is found.

Craig

--
-- --------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------