Subject: Re: counting bits
Posted by condor on Tue, 25 Feb 2003 23:17:05 GMT

View Forum Message <> Reply to Message

JD Smith <jdsmith@as.arizona.edu> wrote in message
news:<pan.2003.02.20.15.43.26.137656.2731@as.arizona.edu>...

```
>  One thing I did notice when creating "random" arrays:
>
>  IDL> print,FORMAT='(F5.2,A)',total(ulong(randomu(sd,100)*2.^31) mod 2 eq 1),$
>            '% odd'
>
>  Try this a few times.  That lowest bit just does not get set.  Some
>  floating-point representation expert must have an explanation.
```

Dunno that this needs an expert: give a /double to the call to rendomu
and it works as expected -- otherwise randomu will return a float
array, floats have 4 byte representation and thus the graininess at
which floats can be represented cannot possibly be better than 1 bit
in 32 (and in reality it's a good bit less).

In other words: you're multiplying floats 0<f<1 with 2.^31 which means
for them to be distinguishable in the last bit the original floats
would have had to have a spacing of 1/2^30 :

```
 m = machar()
 print,m.eps
 1.19209e-07
 print,1/(2^31.)
 4.65661e-10
```

So you have numbers that are at most about 10^7 apart from each other
(the machine precision) and you multiply them with almost 10^10 and
thus will not get numbers that are 'one' apart from each other.

You want weird? Check for all the bits OTHER than the last one:

```
print,FORMAT='(F5.2,A)',total(ulong(randomu(sd,100)*2.^31) and $
 2ul eq 2ul),'% set'

print,FORMAT='(F5.2,A)',total(ulong(randomu(sd,100)*2.^31) and $
 4ul eq 4ul),'% set'

print,FORMAT='(F5.2,A)',total(ulong(randomu(sd,100)*2.^31) and $
 8ul eq 8ul),'% set'
```

etc ...