Subject: Re: counting bits
Posted by JD Smith on Thu, 20 Feb 2003 15:43:26 GMT
View Forum Message <> Reply to Message

On Wed, 19 Feb 2003 08:47:31 -0700, Dick Jackson wrote:

> [sorry for the late entry, it seems my news server has been failing when
> I attach a file to a posting :-(]
>
> "JD Smith" <jdsmith@as.arizona.edu> wrote in message
> news:pan.2003.02.17.23.54.23.693563.14101@as.arizona.edu...
>
>> The method I found fastest
>> (among those I've tried), is a pretty silly and straightforward one,
>> namely table lookup:
>
> I don't think it's silly at all, but taking it one step further will
> really speed it up. If you have enough extra memory for it, just convert
> the whole ULong array to bytes, do one lookup and you're done. One more
> 'gotcha' came up when running this:
>

Good idea, and fast indeed.

> Data:
>
> rand_arr = ULIndgen(2048, 2048)
>
> JD's method:
>
> tot=ulong(total(bits[rand_arr AND 'FF'XUL] + $
> bits[ishft(rand_arr,-8) AND 'FF'XUL]+ $
> bits[ishft(rand_arr,-16) AND 'FF'XUL]+ $
> bits[ishft(rand_arr,-24) AND 'FF'XUL]))
>
> Dick's method:
>
> byte_rand_arr = Byte(rand_arr, 0, N_Elements(rand_arr)*4) tot =
> ULong(Total(bits[byte_rand_arr]))
>
> When I tried this, I got:
>
> IDL> CountingBits
> IShft-AND-lookup method: 2.063 seconds. tot = 46137328
> Byte-lookup method: 0.691 seconds. tot = 46137292
>
> Uh-oh... I set the Total calls to have /Double and then we both get the
> same (I hope correct) answer:

>
> IDL> CountingBits
> IShft-AND-lookup method: 2.063 seconds. tot = 46137344
> Byte-lookup method: 0.671 seconds. tot = 46137344

That's very strange. Here's what I get for my four independent
methods without any /DOUBLE:

```
      3.4187140
   46137344
      6.9928349
   46137344
      1.2564960
   46137344
      1.2767580
   46137344
```

Indeed:

```
IDL> print,ulong(total(bits[rand_arr AND 'FF'XUL] + $
IDL>                bits[ishft(rand_arr,-8)  AND 'FF'XUL]+ $
IDL>                bits[ishft(rand_arr,-16) AND 'FF'XUL]+ $
IDL>                bits[ishft(rand_arr,-24) AND 'FF'XUL], /Double))
   46137344
IDL> print,ulong(total(bits[rand_arr AND 'FF'XUL] + $
IDL>                bits[ishft(rand_arr,-8)  AND 'FF'XUL]+ $
IDL>                bits[ishft(rand_arr,-16) AND 'FF'XUL]+ $
IDL>                bits[ishft(rand_arr,-24) AND 'FF'XUL]))
   46137344
```

I'm not sure what the difference could be (other than Win vs. Linux).

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.

> My kingdom for a /Long flag on Total()!

I'll throw mine in also.

JD