Subject: Re: counting bits
Posted by Dick Jackson on Wed, 19 Feb 2003 15:47:31 GMT
View Forum Message <> Reply to Message

[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:

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
```

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

I attach my test program with handy bonus timer routines TStart and TReport. [Actually copied in-line below, now]

Now, if there were a way to convert to Byte without copying...

Cheers,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

===== CountingBits.pro =====

```
PRO TStart, msg                     ; Timer Start
                                    ; Save current time for use by
TReport
COMMON Timer, t0
IF N_Elements(msg) NE 0 THEN Print, msg
t0 = SysTime(1)
END

PRO TReport, msg                    ; Timer Report
                                    ; Print elapsed time since last
TStart
COMMON Timer, t0
IF N_Elements(msg) EQ 0 THEN msg = ''
Print, Format='(A0, D10.3," seconds.")', msg, SysTime(1)-t0
END

PRO CountingBits

rand_arr = ULIndgen(2048, 2048)

bits = [0, 1, 1, 2, 1, 2, 2, 3, 1, 2, 2, 3, 2, 3, 3, 4, $ ; 0   - 15  */
        1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5, $ ; 16  - 31  */
        1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5, $ ; 32  - 47  */
        2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, $ ; 48  - 63  */
        1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5, $ ; 64  - 79  */
        2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, $ ; 80  - 95  */
        2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, $ ; 96  - 111 */
```

```
      3, 4, 4, 5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7, $ ; 112 - 127 */
      1, 2, 2, 3, 2, 3, 3, 4, 2, 3, 3, 4, 3, 4, 4, 5, $ ; 128 - 143 */
      2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, $ ; 144 - 159 */
      2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, $ ; 160 - 175 */
      3, 4, 4, 5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7, $ ; 176 - 191 */
      2, 3, 3, 4, 3, 4, 4, 5, 3, 4, 4, 5, 4, 5, 5, 6, $ ; 192 - 207 */
      3, 4, 4, 5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7, $ ; 208 - 223 */
      3, 4, 4, 5, 4, 5, 5, 6, 4, 5, 5, 6, 5, 6, 6, 7, $ ; 224 - 239 */
      4, 5, 5, 6, 5, 6, 6, 7, 5, 6, 6, 7, 6, 7, 7, 8  ] ; 240 - 255 */

TStart

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], /Double))

TReport, 'IShft-AND-lookup method: '

Print, 'tot = ', tot

TStart

byte_rand_arr = Byte(rand_arr, 0, N_Elements(rand_arr)*4)
tot = ULong(Total(bits[byte_rand_arr], /Double))

TReport, 'Byte-lookup method: '

Print, 'tot = ', tot

END
```