Subject: Re: bits into bytes
Posted by Heinz Stege on Wed, 01 Aug 2012 23:55:16 GMT
View Forum Message <> Reply to Message

On Wed, 1 Aug 2012 15:49:21 -0700 (PDT), wlandsman wrote:

> I have to write an array of "on-off" states in which the values are stored as bits.   So if I have
>
> x = [1b,0b,1b,1b,1b,0b,1b,1b]
>
> i want to convert this to a single byte value.   (This is the inverse of David Fanning's bitget.pro function.)
> One way to to do this is
>
> IDL> yy = [128b,64b,32b,16b,8b,4b,2b,1b]
> IDL> print,byte(total(yy*x))
> 187b
>
> But there must be a way to do this using masking rather than multiplication. I just can't figure it out right now.   (The conversion will be done for millions of bits).   Thanks, --Wayne

You can use ISHFT instead of the multiplication:

```
  print,total(ishft(x,7b-bindgen(8)),/preserve_type)
```

But I don't expect any advantages on use of a floating point
processor. However the /PRESERVE_TYPE option of the TOTAL function
should speed up the calculation.

Cheers, Heinz

Subject: Re: bits into bytes
Posted by wlandsman on Thu, 02 Aug 2012 03:19:53 GMT
View Forum Message <> Reply to Message

On Wednesday, August 1, 2012 7:55:16 PM UTC-4, Heinz Stege wrote:
>
> processor. However the /PRESERVE_TYPE option of the TOTAL function
>
> should speed up the calculation.
>

Heinz,

Thanks for reminding me of the ISHFT() function and the /PRESERVE_TYPE keyword to TOTAL().   Actually, rather than use TOTAL(), I found the fastest method when dealing with millions of values is to use matrix multiply, as in the following example:

;Create an 8 x n byte array of random 0s and 1s.   This will be compressed by a factor of 8 by storing ;each value in a bit rather than a byte

```
n = 1000000
x = byte(round(randomu(seed,8,n)))

yy = [128b,64b,32b,16b,8b,4b,2b,1b]
return, byte(yy#x)
```

The above code would be slightly faster if there were the equivalent of /PRESERVE_TYPE for matrix multiplication.   Right now -- rather surprisingly, I think -- matrix multiplication of two byte arrays yields a long array.   But the code is very fast anyway.   --Wayne