Subject: Re: 32-bit Unsigned Integers, Was: Unsigned Integers - How?
Posted by James Tappin on Tue, 11 Feb 1997 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
[SNIP]

>
> Now, here is where I start to get unsure of myself. I know
> how to turn *one* unsigned 32-bit integer into its real value.
> You use the BYTE function to individually read the four
> bytes of information in the 32-bit integer and you reconstruct
> those bytes into a DOUBLE-PRECISION value. The code looks
> like this:
>
>   number = data(0)
>   factor = 256.0D
>   realNumber = BYTE(number, 0)*factor^3 + BYTE(number,1)*factor^2 +$
>   BYTE(number,2)*factor^1 + BYTE(number,3)*factor^0
>
> This is for a big endian machine, like most UNIX machines. If
> you are on a little endian machine (like a PC), you will have to
> reverse the order in which the real number is constructed. Your
> code will look like this:
>
>   number = data(0)
>   factor = 256.0D
>   realNumber = BYTE(number, 0)*factor^0 + BYTE(number,1)*factor^1 +$
>   BYTE(number,2)*factor^2 + BYTE(number,3)*factor^3
>
> What I don't know how to do (perhaps Bill Thompson or Mitchell Grunes
> can help us here), is how to do this for the whole array at once in
> an "array" type way. I certainly know how to write a loop! :-)
>
>   realNumbers = DBLARR(N_ELEMENTS(data))
>   factor = 256.0D
>   FOR j=0, N_ELMENTS(data)-1 DO BEGIN
>   realNumbers(j) = BYTE(data(j), 0)*factor^0 + BYTE(data(j),1)*factor^1 +$
>   BYTE(data(j),2)*factor^2 + BYTE(data(j),3)*factor^3
>   ENDFOR
>

Here's my version in vector form

```
IDL> a = '7ffffff0'xl
IDL> print,a$<3>
  2147483632
IDL> x = lindgen(32)+a
```

```
IDL> print,x$<3>
  2147483632  2147483633  2147483634  2147483635  2147483636  2147483637
  2147483638  2147483639  2147483640  2147483641  2147483642  2147483643
  2147483644  2147483645  2147483646  2147483647 -2147483648 -2147483647
 -2147483646 -2147483645 -2147483644 -2147483643 -2147483642 -2147483641
 -2147483640 -2147483639 -2147483638 -2147483637 -2147483636 -2147483635
 -2147483634 -2147483633
IDL> bx = byte(x,0,4,n_elements(x))
IDL> factor = 256.0d0^(3-indgen(4))
IDL> xr = total(bx*factor(*,intarr(n_elements(x))),1)
IDL> print,xr
   2.1474836e+09   2.1474836e+09   2.1474836e+09   2.1474836e+09
   2.1474836e+09   2.1474836e+09   2.1474836e+09   2.1474836e+09
   2.1474836e+09   2.1474836e+09   2.1474836e+09   2.1474836e+09
   2.1474836e+09   2.1474836e+09   2.1474836e+09   2.1474836e+09
   2.1474836e+09   2.1474836e+09   2.1474836e+09   2.1474837e+09
   2.1474837e+09   2.1474837e+09   2.1474837e+09   2.1474837e+09
   2.1474837e+09   2.1474837e+09   2.1474837e+09   2.1474837e+09
   2.1474837e+09   2.1474837e+09   2.1474837e+09   2.1474837e+09
```

For a little-endian machine, factor becomes 256.0d0^indgen(4)

--


```
 +-----------------------+-------------------------------- --+---------+
| James Tappin,          | School of Physics & Space Research |  O__    |
| sjt@star.sr.bham.ac.uk | University of Birmingham       | --  \/` |
| Ph: 0121-414-6462. Fax: 0121-414-3722         |  |
 +-------------------------------------------------------- --+---------+
```