
Subject: Re: 4-bit words

Posted by [orbach](#) on Fri, 22 Dec 1995 08:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

In message <DJxwrH.ou@midway.uchicago.edu> - rivers@cars3.uchicago.edu (Mark Rivers) writes:

:>
:>In article <DJw3nF.D8n@rockyd.rockefeller.edu>, orbach@rockvax.rockefeller.edu (Darren Orbach) writes:

:>>I have a file that consists of a 256*256 array of 4-bit
:>>words created in another application, written as a binary
:>>file. I need to manipulate this array by shifting these
:>>4-bit elements to the right by various amounts, and wrapping
:>>around to the other side of the array. However, since the
:>>smallest data type in WAVE or IDL is a full byte, I don't see
:>>a straightforward way to do this. Any suggestions?

:>
:>I am assuming that your data files have the 4-bit values packed together. If so
:>then the following should create the array you want:

:>
:>; Make a 1-D byte array big enough to hold image from disk
:>IDL> temp = bytarr(2L^15)
:>; Read in the data
:>IDL> readu, lun, temp
:>; Make new array to hold decomposed data
:>IDL> data = bytarr(2, 2L^15)
:>; Low order 4 bits in even elements
:>IDL> data(0,*) = (temp and '00FF'X)
:>; High order 4 bits in odd elements
:>IDL> data(1,*) = (temp/16 and '00FF'X)
:>; Reform into 256x256 array
:>IDL> data = reform(data, 256, 256)

:>

:>

:>Mark Rivers	(312) 702-2279 (office)
:>CARS	(312) 702-9951 (secretary)
:>Univ. of Chicago	(312) 702-5454 (FAX)
:>5640 S. Ellis Ave.	(708) 922-0499 (home)
:>Chicago, IL 60637	rivers@cars3.uchicago.edu (Internet)

:>

Mark's suggestion worked nicely, with the proviso that the hex number to use is '000F'x. The rest of the procedure involved bit-shifting to the right, keeping proper account of where high bits are so formed & moving them separately, and then re-packing the data(2, 2L^15) array back into a temp(1, 2L^15)

array with the operation `temp = (data(0,*) or data(1,*))`.
Thanks to all.

-Darren Orbach
