Subject: Re: Extracting bits with IDL
Posted by chase on Mon, 18 Jul 1994 20:47:36 GMT
View Forum Message <> Reply to Message

>>>> > "dean" == dean <dean@phobos.cira.colostate.edu> writes:

dean>   We are set up to routinely collect GOES-8 GVAR data. Like with
dean> most data from satellites, they put information in the
dean> individual bits. Below is a C structure that extracts the
dean> individual bits from part of the data header which contains some
dean> time information. Extracting individual bits from data with IDL
dean> is difficult. At first I consider building an IDL structure, but
dean> IDL is unable to break down to bits.

dean>   The information can fit into LONARR(2). If anyone has any
dean> suggestion that would allow IDL to extract the information from
dean> these LONARR(2), please forward your comments to me or post
dean> them. I hope to use this information to build a widget to
dean> navigate through this new and fine data set coming from GOES-8.

There was a discussion in this group some time ago about obtaining bit information in IDL. I wrote a very general function for arbitrary base decoding called decode.pro which I include for your use below. One of its applications can be to decode an array of scalars into base 2. For example, with geos = lonarr(2), decode(geos,2) returns an array containing the binary expansion for geos which you can then test.

I am not exactly sure which bits of your LONARR(2) correspond to your C struct definition because the C ANSI standard says that bit-fields are implementation dependent, i.e., non-portable. For example, whether fields are assigned left to right or right to left and/or can overlap word boundaries can vary between machines.

Good luck,
Chris

--- begin included file ---

```
FUNCTION Decode, scl, dim, help=help
;+
; $Id: decode.pro,v 1.1 1994/07/18 16:05:32 chase Exp $
;
; NAME:
;
;   DECODE
;
; PURPOSE:
;
```

```
;   Decode a vector of scalars into the dimensions d1,...,dn
;   in order of increasing significance.
;
; CATEGORY:
;
;   Mathematical Functions
;
; CALLING SEQUENCE:
;
;   Result = DECODE(Scl, Dim)
;
; INPUTS:
;
;   Scl - Vector of scalars to be decoded. Will be converted to
;         integers first, truncating if necessary.
;
;   Dim - If a scalar, then it is converted to an integer base for the
;         decoding, i.e., D1,...,DN=Dim.
;         If a 1 dimensional vector, then it is converted to the
;         integer dimensions D1,...,DN.
;         If > 1 dimensional array, then the dimensions of the array
;         are used for D1,...,DN.
;         The dimensions increase in significance, i.e., the first
;         dimension is the least significant and the last dimension is
;         the most significant.
;
; KEYWORD PARAMETERS:
;
;   HELP - Provide help (this information). No other action is performed.
;
; OUTPUTS:
;
;   Result - Array of size NxM where M is dimension of Scl.
;            Result(*,i) is the decoding of the scalar Scl(i). If
;            Scl(i) is larger then the dimensioned size,
;            i.e. D1x...xDN, then the modulus, Scl(i) mod D1x...xDN,
;            is decoded.  Result(j-1,i) corresponds to dimension Dj
;            with Result(N-1,i) the most significant digit of the
;            decoding.
;
; PROCEDURE:
;
;   Let b1,...,bN be the decoding. Then Scl can be represented as:
;
;   Scl = D1(D2(...(D[N-1]*bN + b[N-1])...+ b3 ) + b2) + b1
;
;   with 0 <= bi < Di
;
```

```
; EXAMPLE:
;
;   scl = [20,63]
;   ; Conversion to base 16
;   print,decode(scl,16)
;
;   ; Conversion to binary (base 2)
;   print,decode(scl,2)
;   ; Invert the decoding
;   print,2^indgen(5)#decode(scl,2)
;
;   ; Arbitrary decoding.  Generates a warning for decoding 63 in
;   ; which case (63 mod 3*4*5) = 3 is decoded.
;   print,decode(scl,[3,4,5])
;   print,[1,3,3*4]#decode(scl,[3,4,5])
;   print,[1,3,3*4,3*4*5]#decode(scl,[3,4,5,6])
;
;   ; Convert 1D index into a multi-dimensional index
;   w=dist(20,20)
;   a=max(w,i)
;   ; Get 2D index for max
;   print,decode(i,w)
;
; MODIFICATION HISTORY:
;
;       Mon Jul 18 15:58:18 1994, Chris Chase S1A <chase@jackson>
;  Fixed/cleaned up.
;
;       Mon Jul 26 12:17:56 1993, Chris Chase <chase@aphill>
;               Created.  Named for similar APL function.
;
;
;-
if keyword_set(help) then begin
   doc_library, 'decode'
   return, ''
endif
s = size(dim)
if (s(0) eq 0) then begin
   d = replicate(long(dim), long(alog(max(scl))/alog(dim)) + 1)
endif else begin
   if (s(0) gt 1) then d = s(1:s(0)) $
   else d = dim
endelse

d = long(d)
nd = n_elements(d)
v = long(scl)
index = lonarr(nd, n_elements(v))
```

```
for i = 0, nd-1 do begin
    f = v/d(i)
    index(i, *) = v-f*d(i)
    v = f
endfor
if (max(v) ne 0) then begin
    print, "Warning - function DECODE: scalar outside dimension " + $
      "bounds, decode of modulus returned."
endif
return, index
end
```

```
--
================================
Bldg 24-E188
The Applied Physics Laboratory
The Johns Hopkins University
(301)953-6000 x8529
chris_chase@jhuapl.edu
```