
Subject: Re: how to do bitwise operation in IDL???

Posted by [wonko](#) on Thu, 24 Jul 1997 07:00:00 GMT

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

jyli@redback.gsfc.nasa.gov (Jason Li) wrote:

> I need to read and set a series of image processing flags which are
> packed into byte size variables. Except the ISHIFT, I could not find
> any other bitwise operators in IDL language.

OR AND NOT do work bitwise. But when I started programming with PV-WAVE,
I didn't even find ISHIFT.

Alex

--

Alex Schuster Wonko@weird.cologne.de PGP Key available
alex@pet.mpin-koeln.mpg.de

Subject: Re: how to do bitwise operation in IDL???

Posted by [thompson](#) on Thu, 24 Jul 1997 07:00:00 GMT

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

jyli@redback.gsfc.nasa.gov (Jason Li) writes:

> Hi,

> I need to read and set a series of image processing flags which are
> packed into byte size variables. Except the ISHIFT, I could not find
> any other bitwise operators in IDL language.

> OK, I can always call C or FORTRAN routines from IDL to accomplish this
> task. But, is there a simpler way to read set bits in IDL ? After all,
> IDL is built on C, isn't it?

As well as the ISHFT function, IDL also contains the bitwise logical operators
AND and OR. For example, if you wanted the value of the 3rd bit from the right
in a byte value, then you could say

```
VALUE = ISHFT(INPUT,-3) AND 1
```

or

```
VALUE = (INPUT AND '8'X) NE 0
```

As a more complicated example, I append a program which extracts 12 bit values
packed into 16 bit integers.

Bill

```
=====
=====
PRO VDS_UNPACK,INPUT,OUTPUT
;+
; Project   : SOHO - CDS
;
; Name      :
; VDS_UNPACK
; Purpose   :
; Unpacks 12-bit VDS values from 16 bit telemetry words.
; Explanation :
; Used to unpack VDS values from the telemetry stream according to one of
; the compression schemes used aboard the CDS instrument. The values of
; OUTPUT are taken from each sequential set of 12 bits in the INPUT
; array. Some of the OUTPUT values will span across consecutive INPUT
; values.
; Use       :
; VDS_UNPACK, INPUT, OUTPUT
; Inputs    :
; INPUT = Input array of packed VDS values. Must be of either byte or
; short integer type.
; Opt. Inputs :
; None.
; Outputs   :
; OUTPUT = Output array of unpacked VDS values. For every three
; elements (six bytes) in the INPUT array, there will be four
; elements in the unpacked array.
; Opt. Outputs:
; None.
; Keywords   :
; None.
; Calls      :
; IEEE_TO_HOST
; Common     :
; None.
; Restrictions:
; None, except those described above for the INPUT array.
; Side effects:
; The OUTPUT array will contain a number of elements equal to a multiple
; of four. If necessary, this will be accomplished by padding the array
; with zeros.
; Category   :
; Data Handling, Telemetry
; Prev. Hist. :
; None.
```

```

; Written   :
; William Thompson, GSFC, October 1992.
; Modified  :
; Version 1, William Thompson, GSFC, October 1992.
; Version 2, William Thompson, GSFC, November 1992.
; Rewrote to use ISHFT.
; Version 3, William Thompson, GSFC, 24 March 1993.
; Allowed INPUT to be either byte or short integer, with an
; arbitrary number of elements. Took host byteordering into
; account.
; Version 4, William Thompson, GSFC, 30 June 1993.
; Added OVERWRITE keyword to REFORM to speed up.
; Version 5, William Thompson, GSFC, 6 July 1995
; Corrected bug where didn't work properly if input array was
; multidimensional.
; Version   :
; Version 5, 6 July 1995
;-
;
;
ON_ERROR,2
;
; Check the number of parameters.
;
IF N_PARAMS() NE 2 THEN MESSAGE, 'Syntax: VDS_UNPACK, INPUT, OUTPUT'
;
; Check the input array.
;
SZ = SIZE(INPUT)
TYPE = SZ(SZ(0)+1)
IF (TYPE NE 1) AND (TYPE NE 2) THEN MESSAGE, $
'INPUT must be either a byte or short integer array'
;
; If INPUT is a byte array, then convert it to short integer type, padding if
; necessary. Take into account the host byteordering.
;
IF TYPE EQ 1 THEN BEGIN
  N = (N_ELEMENTS(INPUT) + 1) / 2
  SHORT = BYTARR(2,N)
  SHORT(0) = INPUT
  SHORT = FIX(SHORT,0,N)
END ELSE SHORT = INPUT
IEEE_TO_HOST, SHORT
;
; Rearrange INPUT into groups of three, padding if necessary, and define the
; output array.
;
N = (N_ELEMENTS(SHORT) + 2) / 3
TEMP = INTARR(3,N)

```

```

TEMP(0) = SHORT(*)
OUTPUT = INTARR(4,N)
;
; Extract the output values from the input array.
;
OUTPUT(0,0) = TEMP(0,*) AND 'FFF'X
OUTPUT(1,0) = (ISHFT(TEMP(0,*), -12) AND 'F'X) OR $
  ISHFT(TEMP(1,*) AND 'FF'X, 4)
OUTPUT(2,0) = (ISHFT(TEMP(1,*), -8) AND 'FF'X) OR $
  ISHFT(TEMP(2,*) AND 'F'X, 8)
OUTPUT(3,0) = ISHFT(TEMP(2,*), -4) AND 'FFF'X
;
; Redefine OUTPUT to be a simple string of numbers.
;
OUTPUT = REFORM(OUTPUT,4*N,/OVERWRITE)
;
RETURN
END

```

Subject: Re: how to do bitwise operation in IDL???

Posted by [Stein Vidar Hagfors H](#) on Fri, 25 Jul 1997 07:00:00 GMT

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

William Thompson wrote:

```

>
> jyli@redback.gsfc.nasa.gov (Jason Li) writes:
>
>> Hi,
>
>> I need to read and set a series of image processing flags which are
>> packed into byte size variables. Except the ISHIFT, I could not find
>> any other bitwise operators in IDL language.
>
>> OK, I can always call C or FORTRAN routines from IDL to accomplish this
>> task. But, is there a simpler way to read set bits in IDL ? After all,
>> IDL is built on C, isn't it?
>
> As well as the ISHFT function, IDL also contains the bitwise logical operators
> AND and OR. For example, if you wanted the value of the 3rd bit from the right
> in a byte value, then you could say
>
>     VALUE = ISHFT(INPUT,-3) AND 1
>
> or
>
>     VALUE = (INPUT AND '8'X) NE 0
>

```

And even though you can implement any logical operation with AND, OR and NOT (I seem to remember that the pairs NOT+AND or NOT+OR will do as well?), there's also an operator for exclusive or - called XOR, as in

```
VALUE = A XOR 1
```

which will flip the least significant bit of A..

Stein Vidar
