Subject: Re: weird behaviour of ishft
Posted by BenHur on Tue, 17 Mar 2009 10:37:19 GMT
View Forum Message <> Reply to Message

Yes, that's probably the reason behind it, thanks. However, I think
the IDL documentation should apply since it's an IDL command and not a
processor op. From the IDL docs:

'Syntax
Result = ISHFT(P1, P2)
Return Value
If P2 is positive, P1 is left shifted P2 bit positions with 0 bits
filling vacated positions.'

There is no limit mentioned for P2...

Also, the IDL implementation is inconsistent: for 64bit types the
limit does not exist and
ishft(long64(1),64) = 0 as expected, for all other types there's the 0
to 31 limit. Then again, 64bit types aren't even mentioned in the
ishft documentation...

So, my guess: it's just crappy documentation.

Cheers, Ben

On 17 Mrz., 10:46, FÖLDY Lajos <fo...@rmki.kfki.hu> wrote:
> On Tue, 17 Mar 2009, BenHur wrote:
>> Hi everyone,
>
>> Can somebody explain to me, why for certain types ishft shifts only by
>> modulo 32 bits?
>> Example:
>> DISR> print,ishft(1,[31,32])
>> 0 1
>> DISR> print,!version
>> { x86 Win32 Windows Microsoft Windows 6.4 Mar 23 2007 32 64}
>
>> I would expect ishft(1,32) = 0. Am I missing something? Is it my
>> outdated IDL version or does it happen with newer versions/other
>> platforms?
>
>> Cheers, Ben
>
> It's normal for the right shift (SHR) op on x86 processors. From the Intel
> manual: 'The count is masked to 5 bits, which limits the count range to 0
> to 31.'

>
> regards,
> lajos