

Re: Can someone please explain – real numbers

Re: Can someone please explain – real numbers

Source: <http://coding.derkeiler.com/Archive/Delphi/comp.lang.pascal.delphi.misc/2006-11/msg00142.html>

- *From:* "Maarten Wiltink" <maarten@xxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Thu, 9 Nov 2006 10:57:13 +0100
-

"John Dough" <nobody@xxxxxxxxxxxxxxxxxxxxx> wrote in message <news:885512lt9g8r5b68gt8i7713nktgg9prrl@xxxxxxxxxxx>

On Wed, 08 Nov 2006 20:51:34 GMT, Chris Cheney
<XXChris.CheneyXX@xxxxxxxxxxx> wrote:

Note that, for example, the simple decimal fraction 0.1 is the recurring binary fraction 0.000110011001100110011001100110 (0110 recurring) and cannot therefore be stored precisely as a single/double/extended/real.

The FPU knows how to handle repeating fractions, regardless of their base. The repeating portion is truncated (and rounded, if necessary) at the last available digit of precision for any given float type. In the case of "extended", the decimal portion terminates at the 18th position.

It's rather generous to see 'truncating' and say 'handling'.

I'm not sure what leads you to believe that 0.1 cannot be stored precisely.

The simple fact that not all the bits in it are preserved.

Did you try it with code? I did...and it works fine.
I can store 0.1, and I get back 0.1
If you get back what you store, then by all accounts, it's precise.

It took some work, but here's a program that demonstrates the problem:

Re: Can someone please explain – real numbers

Re: Can someone please explain – real numbers

```
var X: Real;  
begin  
X:=0.2; { To start the recurring fraction immediately }  
while True do begin  
X:=16*X;  
X:=X-3;  
end;  
end.
```

Sixteen times 0.2 equals 3.2, subtracting 3 should leave us with the original 0.2. Ad infinitum. Right?

Not in finite precision. Looking at the finite bitstring above (minus the first zero after the p