

## Re: "Shifting" floating point numbers

---

*Source:* [http://coding.derkeiler.com/Archive/C\\_CPP/comp.lang.c/2006-03/msg03404.html](http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2006-03/msg03404.html)

---

- *From:* "santosh" <[santosh.k83@xxxxxxxxxx](mailto:santosh.k83@xxxxxxxxxx)>
  - *Date:* 21 Mar 2006 10:53:57 -0800
- 

woess...@xxxxxxxxxx wrote:

Does anyone know of a fast way to multiply floating point numbers by powers of two? Conceptually, all you need to do is add to the mantissa. But can I write C code (or x86 assembly) to accomplish this without a full-blown multiply?

For example, I'd like to be able to do the following very quickly:

```
double x;
double y;

x = 42.13;
y = (1 << 9) * x;
```

If you know the floating point representation of your implementation and underlying hardware, you can copy the float value to an unsigned long variable, shift the appropriate bits and copy the value back to the float variable. All this implies that you'll have to know details about your implementation and thus risk losing some portability.

These days, FPU's are quite fast, so a direct multiply on the float should be efficient enough unless your code is time critical.

.