

## Re: Bitwise OR and AND without bitwise operators....

---

*Source:* <http://sci.tech-archive.net/Archive/sci.math/2006-02/msg04545.html>

---

- *From:* [stush@xxxxxxxxxxxxxxxx](mailto:stush@xxxxxxxxxxxxxxxx)
  - *Date:* 28 Feb 2006 11:16:24 -0800
- 

CelticDaddio wrote:

I am working in a programming language (GLSL) which does not support bitwise operators, but I need to perform the following types of calculations:

```
float b = (float)((counts & maskb) >> offset) / gbitres;
```

I can do the bit shifts by

Bit Shift Right ...  $i \gg N = i / 2^N$ ,

Bit Shift Left...  $i \ll N = i * 2^N$ ,

right, but how can I do the bitwise AND? or a bitwise OR?

the lsb of n would be  $n - ((n/2)*2)$ .

the 2nd lsb of n would be  $m = n/2; m - ((m/2)*2)$

etc.

Notice you can re-use some results if you work lsb to msb.

So just iterate bitwise yourself and add  $2^k$  to your result when the bitwise operation on the kth bit is 1.

I don't know if this is the fastest way, but without bitwise arithmetic support (which is why I assume bitwise operations are reserved), I would think any solution would have to iterate over the operand's bits.

.