

Notes

There seems to be a problem with `exp(double)` and our emulator. I haven't been able to track it down yet. This does not occur with the emulator supplied by Russell King.

I also found one oddity in the emulator. I don't think it is serious but will point it out. The ARM calling conventions require floating point registers `f4-f7` to be preserved over a function call. The compiler quite often uses an `stfe` instruction to save `f4` on the stack upon entry to a function, and an `ldfe` instruction to restore it before returning.

I was looking at some code, that calculated a double result, stored it in `f4` then made a function call. Upon return from the function call the number in `f4` had been converted to an extended value in the emulator.

This is a side effect of the `stfe` instruction. The double in `f4` had to be converted to extended, then stored. If an `lfn/sfm` combination had been used, then no conversion would occur. This has performance considerations. The result from the function call and `f4` were used in a multiplication. If the emulator sees a multiply of a double and extended, it promotes the double to extended, then does the multiply in extended precision.

This code will cause this problem:

```
double x, y, z; z = log(x)/log(y);
```

The result of `log(x)` (a double) will be calculated, returned in `f0`, then moved to `f4` to preserve it over the `log(y)` call. The division will be done in extended precision, due to the `stfe` instruction used to save `f4` in `log(y)`.