Notes on register bank usage in the kernel

Introduction

The SH-3 and SH-4 CPU families traditionally include a single partial register bank (selected by SR.RB, only r0 ... r7 are banked), whereas other families may have more full-featured banking or simply no such capabilities at all.

SR.RB banking

In the case of this type of banking, banked registers are mapped directly to r0 ... r7 if SR.RB is set to the bank we are interested in, otherwise ldc/stc can still be used to reference the banked registers (as r0_bank ... r7_bank) when in the context of another bank. The developer must keep the SR.RB value in mind when writing code that utilizes these banked registers, for obvious reasons. Userspace is also not able to poke at the bank1 values, so these can be used rather effectively as scratch registers by the kernel.

Presently the kernel uses several of these registers.

- r0_bank, r1_bank (referenced as k0 and k1, used for scratch registers when doing exception handling).
- r2_bank (used to track the EXPEVT/INTEVT code)
    - Used by do_IRQ() and friends for doing irq mapping based off of the interrupt exception vector jump table offset
- r6_bank (global interrupt mask)
    - The SR.IMASK interrupt handler makes use of this to set the interrupt priority level (used by local_irq_enable())
- r7_bank (current)