****
    4937 Sat Feb 7 18:57:52 2015
new/usr/src/uts/armv6/ml/glocore.s
armv6: simplify highvecs enabling code
Use the barrel shifter, Luke.
****

```
/*
 * This file and its contents are supplied under the terms of the
 * Common Development and Distribution License ("CDDL"), version 1.0.
 * You may only use this file in accordance with the terms of version
 * 1.0 of the CDDL.
 *
 * A full copy of the text of the CDDL should have accompanied this
 * source.  A copy of the CDDL is also available via the Internet at
 * http://www.illumos.org/license/CDDL.
 */

/*
 * Copyright 2013 (c) Joyent, Inc. All rights reserved.
 * Copyright (c) 2015 Josef 'Jeff' Sipek <jeffpc@josefsipek.net>
 */

#include <sys/asm_linkage.h>
#include <sys/machparam.h>
#include <sys/cpu_asm.h>

#include "assym.h"

/*
 * Every story needs a beginning. This is ours.
 */

/*
 * Each of the different machines has its own locore.s to take care of getting
 * the machine specific setup done.  Just before jumping into fakebop the
 * first time, we call this machine specific code.
 */

/*
 * We are in a primordial world here. The loader is going to come along and
 * boot us at _start. As we've started the world, we also need to set up a
 * few things about us, for example our stack pointer. To help us out, it's
 * useful to remember what the loader set up for us:
 *
 * - unaligned access are allowed (A = 0, U = 1)
 * - virtual memory is enabled
 *   - we (unix) are mapped right were we want to be
 *   - a UART has been enabled & any memory mapped registers have been 1:1
 *     mapped
 *   - ATAGs have been updated to tell us what the mappings are
 * - I/D L1 caches have been enabled
 */

        /*
         * External globals
         */
        .globl  _locore_start
        .globl  mlsetup
        .globl  sysp
        .globl  bootops
        .globl  bootopsp
        .globl  t0

        .data
        .comm   t0stack, DEFAULTSTKSZ, 32
        .comm   t0, 4094, 32
```

```
/*
 * Recall that _start is the traditional entry point for an ELF binary.
 */
        ENTRY(_start)
        ldr     sp, =t0stack
        ldr     r4, =DEFAULTSTKSZ
        add     sp, r4
        bic     sp, sp, #0xff

        /*
         * establish bogus stacks for exceptional CPU states, our exception
         * code should never make use of these, and we want loud and violent
         * failure should we accidentally try.
         */
        cps     #(CPU_MODE_UND)
        mov     sp, #-1
        cps     #(CPU_MODE_ABT)
        mov     sp, #-1
        cps     #(CPU_MODE_FIQ)
        mov     sp, #-1
        cps     #(CPU_MODE_IRQ)
        mov     sp, #-1
        cps     #(CPU_MODE_SVC)

        /* Enable highvecs (moves the base of the exception vector) */
        mrc     p15, 0, r3, c1, c0, 0
        orr     r3, r3, #(1 << 13)
        mov     r4, #1
        lsl     r4, r4, #13
        orr     r3, r3, r4
        mcr     p15, 0, r3, c1, c0, 0

        /* invoke machine specific setup */
        bl      _mach_start

        bl      _fakebop_start
        SET_SIZE(_start)
```
______***unchanged_portion_omitted_***