```
asmlinkage __visible void __init start_kernel(void)
{
......

        ftrace_init();

    dump_p15();

        /* Do the rest non-__init'ed, we're now alive */
        rest_init();
}

static void dump_p15(void)
{
    unsigned int tmp = 0;

    // dump ID register
    asm(
        "mrc p15, 0, %0, c0, c0, 0\n"
        :"=r"(tmp)
        :
    );
    printk("p15-c0: 0x%8x  ", tmp);

    tmp = 0;

    // dump cache type register
    asm(
        "mrc p15, 0, %0, c0, c0, 1\n"
        :"=r"(tmp)
        :
    );
    printk("0x%8x  ", tmp);

    tmp = 0;

    // dump TLB Type register
    asm(
        "mrc p15, 0, %0, c0, c0, 3\n"
        :"=r"(tmp)
        :
    );
    printk("0x%8x  ", tmp);

    tmp = 0;

    // dump MPIDR register
    asm(
        "mrc p15, 0, %0, c0, c0, 5\n"
        :"=r"(tmp)
        :
    );
```

```
        printk("0x%8x  \n", tmp);

        tmp = 0;

        // dump c1 register
        asm(
            "mrc p15, 0, %0, c1, c0, 0\n"
            :"=r"(tmp)
            :
        );
        printk("p15-c1: 0x%8x  \n", tmp);

        tmp = 0;

        // dump c2 register, Translation Table Base (TTB) register
        asm(
            "mrc p15, 0, %0, c2, c0, 0\n"
            :"=r"(tmp)
            :
        );
        printk("p15-c2: 0x%8x  \n", tmp);

        tmp = 0;

        // dump c3 register, 16 domains
        asm(
            "mrc p15, 0, %0, c3, c0, 0\n"
            :"=r"(tmp)
            :
        );
        printk("p15-c3: 0x%8x  \n", tmp);

        tmp = 0;

        // dump c4 register, reserved, could not read !!!
    //    asm(
    //        "mrc p15, 0, %0, c4, c0, 0\n"
    //        :"=r"(tmp)
    //        :
    //    );
    //    printk("p15-c4(reserved): 0x%8x  \n", tmp);
    //
    //    tmp = 0;

        // dump c5 register, data invalidation status register
        asm(
            "mrc p15, 0, %0, c5, c0, 0\n"
            :"=r"(tmp)
            :
        );
        printk("p15-c5: 0x%8x  ", tmp);

        tmp = 0;

        // dump c5 register, instruction invalidation status register
        asm(
            "mrc p15, 0, %0, c5, c0, 1\n"
            :"=r"(tmp)
            :
        );
        printk("0x%8x  \n", tmp);

        tmp = 0;

        // dump c6 register, data invalidation address
        asm(
            "mrc p15, 0, %0, c6, c0, 0\n"
            :"=r"(tmp)
            :
        );
        printk("p15-c6: 0x%8x  ", tmp);

        tmp = 0;

        // dump c6 register, instruction invalidation address
        asm(
            "mrc p15, 0, %0, c6, c0, 2\n"
            :"=r"(tmp)
            :
        );
        printk("0x%8x  \n", tmp);

        tmp = 0;

        // c7, c8 are write-only
        // c7 is for controling cache
        // c8 is for clearing TLB entries

        // dump c12 register, vector address
        asm(
            "mrc p15, 0, %0, c12, c0, 0\n"
            :"=r"(tmp)
            :
        );
        printk("p15-c12: 0x8%x  ", tmp);

        // dump c13 register, fast context switch
    }
```

output from Gemstone2 board

```
p15-c0: 0x410fd034  0x84448004  0x       0  0x80ffff00
```

p15-c1: 0x30c5383d

p15-c2: 0x   3000

p15-c3: 0xfffffffd

p15-c5: 0x57731fe9 0x   163e

p15-c6: 0x7f7b2fdf 0x7b6bffdd

p15-c12: 0x83fe9f000

c0 - id register is 0x410fd034

0100,0001,0000,1111,1101,0000,0011,**0100**

0100 - marvell定义的处理器版本号

0100,0001,0000,1111,**1101,0000,0011**,0100

1101,0000,0011 - marvell定义的产品编号

0100,0001,0000,**1111**,1101,0000,0011,0100

1111 - ARM arch version

0100,0001,0000,1111,1101,0000,0011,0100

生产厂商编号，marvell为0x41

---

c0 - MPIDR register is 0x80ffff00

c1 - control register

0x30c5383d

0011,0000,1100,0101,0011,1000,0011,1101

0011,0000,1100,0101,0011,1000,0011,110**1**

0: disable MMU

1: enable MMU

0011,0000,1100,0101,0011,1000,0011,11**0**1

0: disable address alignment

1: enable address alignment

0011,0000,1100,0101,0011,1000,0011,1**1**01

if data cache and instruction cache is different

0: disable data cache

1: enable data cache

if data cache and instruction cache share the same cache

0: disable whole cache

1: ensable whole cache

0011,0000,1100,0101,0011,1000,0011,**1**101

0: disbale write buffer

1: enable write buffer

0011,0000,1100,0101,0011,1000,0011,1101

0011,0000,1100,0101,0011,1000,0011,1101

Gemstone2 doesn't support 26-bit address, so return "1"

0011,0000,1100,0101,0011,1000,0011,1101

0: little endian

1: big endian

0011,0000,1100,0101,0011,1000,0011,1101

0: diable跳转预测

1： enable跳转预测

0011,0000,1100,0101,0011,1000,0011,1101

when data cache and instruction cache are different

0: disable instruction cache

1: enable instruction cache

0011,0000,1100,0101,0011,1000,0011,1101

control vector table location

0: low vector, 0 - 0x1c

1: high vector, 0xffff0000 - 0xffff001c

0011,0000,1100,0101,0011,1000,0011,1101

control cache中的淘汰算法

0: 常规的cache淘汰算法, 如随机淘汰

1：预测性cache淘汰算法, 如round-robin

0011,0000,1100,0101,**0**011,1000,0011,1101

0: 保持ARMv5以上版本的正常功能

1：将ARMv5以上版本与以前版本兼容。即不跟据跳转地址的bit[0]进行ARM和 Thumb状态切换。bit[0] = 0, ARM instruction, bit[0] = 1, Thumb instruction

---

c2 - Page Directory Table address

0x　3000

页表的物理地址在12K处。

---

p15-c3: 0xfffffffd

1111,1111,1111,1111,1111,1111,1111,11**01**

D0 domain = 01

00:当前级别下,该内存区域不允许被访问,任何的访问都会引起一个domain fault,这时 AP位无效

01:当前级别下,该内存区域的访问必须配合该内存区域的段描述符中AP位进行权检查

10:保留状态(我们最好不要填写该值,以免引起不能确定的问题)

11:当前级别下,对该内存区域的访问都不进行权限检查。 这时 AP位无效

---

p15-c5: 0x57731fe9 0x   163e

0x57731fe9,数据失效 status

0x   163e,instruction 失效 status

bit 0 - bit 3 (4 bits),status id

引起访问失效的原因	状态标识	域标识	C6
终端异常(Terminal Exception)	0010 (0x2)	无效	生产商定义
中断向量访问异常(Vector Exception)	0000 (0x0)	无效	有效
地址对齐	00x1	无效	有效
一级页表访问失效	1100 (0xc)	无效	有效
二级页表访问失效	1110 (0xe)	有效	有效
基于段的地址变换失效	0101 (0x5)	无效	有效
基于页的地址变换失效	0111 (0x7)	有效	有效
基于段的存储访问中域控制失效	1001 (0x9)	有效	有效
基于页的存储访问中域控制失效	1101 (0xd)	有效	有效
基于段的存储访问中访问权限控制失效	1111 (0xf)	有效	有效
基于页的存储访问中访问权限控制失效	0100 (0x4)	有效	有效
基于段的 cache 预 取时外部存储系统失效	0110 (0x6)	有效	有效

基于页的 cache 预 取时外部存储系统失效	1000 (0x8)	有效	有效
基于段的非 cache 预取时外部存储系统失效	1010 (0xa)	有效	有效

bit 4 - bit 7, domain id

---

p15-c6: 0x7f7b2fdf 0x7b6bffdd

0x7f7b2fdf - 访问data失效address (virtual address)

0x7b6bffdd - 访问instruction失效address (virtual address)

---

p15-c12: 0x83fe9f000

设置异常vector base address

mcr p15, 0, Rd, c12, c0, 0

bit 0 - bit 4, reserve

bit 5 - bit 31, exception vector base address

Rd = exception vector base address