in drivers/platform/pegmatite/ciu.c

```
1.  int ciu_setup_boot(unsigned int cpu, unsigned long address)
2.  {
3.          u32 __iomem *core_cfg_addr;
4.          u32 __iomem *core_boot_addr_high;
5.          u32 __iomem *core_boot_addr_low;
6.          u32 arch_mode_mask;
7.          u64 addr64 = (u64)address; /* Address may be up to 64 bits */
8.          int val;
9.
10.         if (!ciu_base) {
11.                 pr_warn("Can't boot CPU. CIU is uninitialized\n");
12.                 return 1;
13.         }
14.
15.         /* Select the correct CIU configuration control register for the requeste
    d cpu */
16.         switch(cpu) {
17.                 case 0:
18.                         core_cfg_addr = ciu_base + A53_CORE_0_CFG_CTRL;
19.                         core_boot_addr_low  = ciu_base + A53_CORE_0_RVBARADDRL;
20.                         core_boot_addr_high = ciu_base + A53_CORE_0_RVBARADDRH;
21.                         arch_mode_mask = A53_CORE_01_CFG_CTRL_ARCH_MODE_MASK;
22.                         break;
23.                 case 1:
24.                         core_cfg_addr = ciu_base + A53_CORE_1_CFG_CTRL;
                            ①
25.                         core_boot_addr_low  = ciu_base + A53_CORE_1_RVBARADDRL;
                    ②
26.                         core_boot_addr_high = ciu_base + A53_CORE_1_RVBARADDRH;
                    ③
27.                         arch_mode_mask = A53_CORE_01_CFG_CTRL_ARCH_MODE_MASK;

28.                         break;
29.                 case 2:
30.                         core_cfg_addr = ciu_base + A53_CORE_2_CFG_CTRL;
31.                         core_boot_addr_low  = ciu_base + A53_CORE_2_RVBARADDRL;
32.                         core_boot_addr_high = ciu_base + A53_CORE_2_RVBARADDRH;
33.                         arch_mode_mask = A53_CORE_23_CFG_CTRL_ARCH_MODE_MASK;
34.                         break;
35.                 case 3:
36.                         core_cfg_addr = ciu_base + A53_CORE_3_CFG_CTRL;
37.                         core_boot_addr_low  = ciu_base + A53_CORE_3_RVBARADDRL;
38.                         core_boot_addr_high = ciu_base + A53_CORE_3_RVBARADDRH;
39.                         arch_mode_mask = A53_CORE_23_CFG_CTRL_ARCH_MODE_MASK;
40.                         break;
41.                 default:
42.                         printk(KERN_ERR "%s: invalid cpu for secondary boot %d\n"
    , __func__, cpu);
43.                         return -ENODEV;
44.         }
45.
46.         val = readl(core_cfg_addr);
47.
```

```
48.          /* Set VINITHI, this makes the reset vector in the SQU */
49.          val |= A53_CORE_X_CFG_CTRL_VINITHI_MASK;
50.
51.          /* Write the address. Only has effect for ARMv8 */
52.          writel(addr64 >> 2, core_boot_addr_low);
53.          writel(addr64 >> (32+2), core_boot_addr_high);
54.
55.  #ifdef CONFIG_ARM
56.          /* Set the boot control register to SQU bank 0, 0xd1000000 */
57.          if (!pegmatite_is_fpga()) {
58.                  if(is_gr2_reva()) {
59.                          writel(REVA_BOOT_FROM_D1000000, ciu_base + BOOT_CTRL);
60.                  } else {
61.                          writel(REVB_BOOT_FROM_D1000000, ciu_base + BOOT_CTRL);
          ④
62.                  }
63.          }
64.          /* Make sure arch mode is not set so the core boots in aarch32 */
65.          val &= ~arch_mode_mask;
66.  #else
67.          val |= arch_mode_mask;
68.  #endif
69.          writel(val, core_cfg_addr);
70.
71.          return 0;
72.  }
```

①

The offset of Coretext A53 Core 1 Configuration Control Register

bit 27：0 - AArch32 mode, 1 - AArch64 mode

bit 26: Noninvasive debug enable, 0 = disable 1 = enable

.....

bit 8: 0 - little endian, 1 - big endian

bit 7: Thumb except init, 0 = disable, 1 = enable

bit 5: CP15 disable, 0 = disable, 1 = enable

bit 0: VINITHI, 0 - disable, 1 = enable

②③

core 1's RVBARADDR,即core 1 reset以后从哪儿开始运行。但只对ARMv8有效，所以在G2 LSP中没用。

G2 LSP工作在AArch32 mode。

④

设置当core 1 reset以后从0xd100,0000开始。而pegmatite_smp_jump()就位于该处。