```
1.  int __cpu_up(unsigned int cpu, struct task_struct *idle)
2.  {
3.          int ret;
4.
5.          if (!smp_ops.smp_boot_secondary)
6.                  return -ENOSYS;
7.
8.          /*
9.           * We need to tell the secondary core where to find
10.          * its stack and the page tables.
11.          */
12.         secondary_data.stack = task_stack_page(idle) + THREAD_START_SP;
13. #ifdef CONFIG_ARM_MPU
14.         secondary_data.mpu_rgn_szr = mpu_rgn_info.rgns[MPU_RAM_REGION].drsr;
15. #endif
16.
17. #ifdef CONFIG_MMU
                ①
18.         secondary_data.pgdir = get_arch_pgd(idmap_pgd);
19.         secondary_data.swapper_pg_dir = get_arch_pgd(swapper_pg_dir);
20. #endif
21.         sync_cache_w(&secondary_data);
22.
23.         /*
24.          * Now bring the CPU into our world.
25.          */
26.         ret = smp_ops.smp_boot_secondary(cpu, idle);                    ②
27.         if (ret == 0) {
28.                 /*
29.                  * CPU was successfully started, wait for it
30.                  * to come online or time out.
31.                  */
32.                 wait_for_completion_timeout(&cpu_running,                       ③
33.                                             msecs_to_jiffies(1000));
34.
35.                 if (!cpu_online(cpu)) {
36.                         pr_crit("CPU%u: failed to come online\n", cpu);
37.                         ret = -EIO;
38.                 }
39.         } else {
40.                 pr_err("CPU%u: failed to boot: %d\n", cpu, ret);
41.         }
42.
43.
44.         memset(&secondary_data, 0, sizeof(secondary_data));
45.         return ret;
46. }
```

①

每个core有自己独立的register和co-processor，但virtual memory是share的。这里把代表virtual memory的信息复制到secondary_data variable中。

在arch/arm/kernel/head.S中的secondary_startup() function中会用到。

```
ENTRY(secondary_startup)

    ......

    /*
     * Use the page tables supplied from  __cpu_up.
     */
    adr   r4, __secondary_data
    ldmia      r4, {r5, r7, r12}        @ address to jump to after
    sub  lr, r4, r5                 @ mmu has been enabled
    ldr    r4, [r7, lr]             @ get secondary_data.pgdir
    add  r7, r7, #4
    ldr    r8, [r7, lr]             @ get secondary_data.swapper_pg_dir
    adr   lr, BSYM(__enable_mmu)            @ return address
    mov r13, r12                    @ __secondary_switched address
 ARM(      add  pc, r10, #PROCINFO_INITFUNC   ) @ initialise processor

    ......
```

ENDPROC(secondary_startup)

.type__secondary_data, %object

__secondary_data:

.long.

.longsecondary_data

②

这里运行的是drivers/platform/pegmatite/smp/platsmp.c中的

```
1.  static int pegmatite_boot_secondary(unsigned int cpu, struct task_struct *idle)
2.  {
3.          return __pegmatite_boot_secondary(cpu);
4.  }
5.
6.  static int __pegmatite_boot_secondary(unsigned int cpu)
7.  {
8.          pegmatite_boot_cpus_to(cpumask_of(cpu), virt_to_phys(&pegmatite_secondary
    _startup));
9.
10.         return 0;
11. }
```

这里的pegmatite_secondary_startup是定义在headsmp.S中的function

ENTRY(pegmatite_secondary_startup)

pegmatite_cpu_prepare

b secondary_startup

ENDPROC(pegmatite_secondary_startup)

```
1.  /*
2.   * Boot a CPU to "address"
3.   */
4.  void pegmatite_boot_cpus_to(const cpumask_t *cpus, unsigned long address)
5.  {
6.          int cpu;
7.          /*
8.           * For ARMv7, we have to patch a table which will be read from code
9.           * that executes from the reset vector which is configured to SRAM.
10.          * ARMv8 is much simpler. We just put the address in the CIU
11.          */
12. #ifdef CONFIG_ARM
13.
14.         if (pegmatite_is_fpga()) {
15.                 int *a53address_addr;
16.                 a53address_addr = ioremap_nocache(0xfd00006c, 4);
17.                 writel(0x00000003, a53address_addr);
18.                 iounmap(a53address_addr);
19.         }
20.
21.         for_each_cpu(cpu, cpus)
22.                 writel(address, &pegmatite_boot_addr[cpu]);                    (
    A)
23.         __cpuc_clean_dcache_area((void *)pegmatite_boot_addr, sizeof(pegmatite_bo
    ot_addr[0]) * num_present_cpus()); (B)
24. #endif
25.         /* Configure the CIU so the core's reset vector is the SQU */
26.         for_each_cpu(cpu, cpus)
27.                 ciu_setup_boot(cpu, address);
                    (C)
28.
29.         /* Bring the core out of reset */
30.         pegmatite_boot_cpus(cpus);
                    (D)
31. }
```

(A)

pegmatite_boot_addr[1] = pegmatite_secondary_startup

(B)

flush cache

(C)

设置CIU register,使得secondary core从特定地址启动(而不是默认的0地址或0xffff,0000开始)

(D)

给secondary core加power和clock.

③

通过cpu_running来等待core 1的启动完成。

释放cpu_running是在secondary_start_kernel() in arch/arm/kernel/smp.c

```
1.  /*
2.   * This is the secondary CPU boot entry.  We're using this CPUs
3.   * idle thread stack, but a set of temporary page tables.
4.   */
5.  asmlinkage void secondary_start_kernel(void)
6.  {
7.          struct mm_struct *mm = &init_mm;
8.          unsigned int cpu;
9.
10.         /*
11.          * The identity mapping is uncached (strongly ordered), so
12.          * switch away from it before attempting any exclusive accesses.
13.          */
14.         cpu_switch_mm(mm->pgd, mm);
15.         local_flush_bp_all();
16.         enter_lazy_tlb(mm, current);
17.         local_flush_tlb_all();
18.
19.         /*
20.          * All kernel threads share the same mm context; grab a
21.          * reference and switch to it.
22.          */
23.         cpu = smp_processor_id();
24.         atomic_inc(&mm->mm_count);
25.         current->active_mm = mm;
26.         cpumask_set_cpu(cpu, mm_cpumask(mm));
27.
28.         cpu_init();
29.
30.         printk("CPU%u: Booted secondary processor\n", cpu);
31.
32.         preempt_disable();
33.         trace_hardirqs_off();
34.
35.         /*
36.          * Give the platform a chance to do its own initialisation.
37.          */
38.         if (smp_ops.smp_secondary_init)
39.                 smp_ops.smp_secondary_init(cpu);
40.
41.         notify_cpu_starting(cpu);
42.
43.         calibrate_delay();
44.
45.         smp_store_cpu_info(cpu);
46.
47.         /*
48.          * OK, now it's safe to let the boot CPU continue.  Wait for
49.          * the CPU migration code to notice that the CPU is online
50.          * before we continue - which happens after __cpu_up returns.
51.          */
52.         set_cpu_online(cpu, true);
53.         complete(&cpu_running);            <---- Note
```

```
54.
55.             local_fiq_enable();
56.             local_irq_enable();
57.
58.             /*
59.              * OK, it's off to the idle thread for us
60.              */
61.             cpu_startup_entry(CPUHP_ONLINE);
62.     }
```

Note：

Secondary core的初始化过程不能太耽误时间，如果在运行到释放cpu_running以前耗时超过1 second（1000 msecs）,则该core就会被认为

启动失败了。