

对multi-core而言，不存在enable / disable所有core的interrupt的API.只有enable / disable 当前core的API.

for example:

```
1. void kgdb_roundup_cpus(unsigned long flags)
2. {
3.     local_irq_enable();
4.     smp_call_function(kgdb_call_nmi_hook, NULL, 0);
5.     local_irq_disable();
6. }
```

```
1. #include <linux/irqflags.h>
```

```
1. #define local_irq_enable() do { raw_local_irq_enable(); } while (0)
2. #define local_irq_disable() do { raw_local_irq_disable(); } while (0)
3.
4. #define raw_local_irq_disable() arch_local_irq_disable()
5. #define raw_local_irq_enable() arch_local_irq_enable()
```

显然enable / disable interrupt是arch specific的。

```
1. #if __LINUX_ARM_ARCH__ >= 6
2. static inline void arch_local_irq_enable(void)
3. {
4.     asm volatile(
5.         "    cpsie i          @ arch_local_irq_enable"
6.         :
7.         :
8.         : "memory", "cc");
9. }
10.
11. static inline void arch_local_irq_disable(void)
12. {
13.     asm volatile(
14.         "    cpsid i          @ arch_local_irq_disable"
15.         :
16.         :
17.         : "memory", "cc");
18. }
19.
20. #else
21.     .....
22. #endif
```

instruction	functionality
cpsid i	disable interrupt
cpsie i	enable interrupt
cpsid f	disable exception
cpsie f	enable exception

在explicit enable / disable interrupt时要考虑原来的状态。

```
1. unsigned long flags;  
2.  
3. local_irq_save(flags)  
4.  
5. ....  
6.  
7. local_irq_restore(flags)
```