



# [12] 发明专利申请公布说明书

[21] 申请号 200810067040.2

[43] 公开日 2008年9月17日

[11] 公开号 CN 101266561A

[22] 申请日 2008.4.29

[21] 申请号 200810067040.2

[71] 申请人 中兴通讯股份有限公司

地址 518057 广东省深圳市南山区高新技术产业园科技南路中兴通讯大厦法务部

[72] 发明人 赵阳 过仕好

[74] 专利代理机构 深圳市君胜知识产权代理事务所  
代理人 杨宏

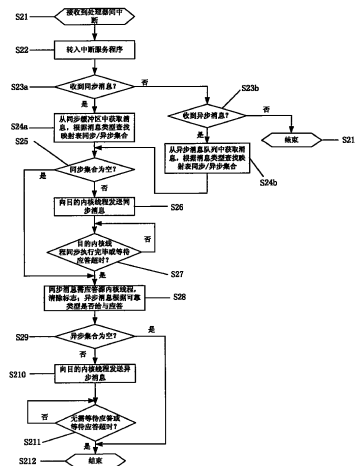
权利要求书 3 页 说明书 11 页 附图 5 页

## [54] 发明名称

一种在多核多线程处理器中的核间消息通信方法

## [57] 摘要

本发明公开了一种在多核多线程处理器中的核间消息通信方法，其包括以下步骤：确立各内核线程在各种消息类型触发下的同步或异步关系，建立各内核线程之间异步消息队列和同步消息缓存；源内核线程读取当前消息所对应的内核线程同步集合或异步集合，将封装的消息体放入所述异步消息队列和同步消息缓存中；目的内核线程收到处理器间中断后，从所述异步消息队列和同步消息缓存中解析消息内容并执行完成操作。由于采用了多核多线程处理器的处理器间中断封装的同步/异步消息通信模式，保证了各内核线程可在不同应用场景中操作执行正确的相对关系，满足了某些关键的应用平台的开发需要，极大提高了多核多线程系统运行的可靠性和灵活性。



1、一种在多核多线程处理器中的核间消息通信方法，其包括以下步骤：

A、确立各内核线程在各种消息类型触发下的同步或异步关系，建立各内核线程之间异步消息队列和同步消息缓存；

B、源内核线程读取当前消息所对应的内核线程同步集合或异步集合，将封装的消息体放入所述异步消息队列和同步消息缓存中；

C、目的内核线程收到处理器间中断后，从所述异步消息队列和同步消息缓存中解析消息内容并执行完成操作。

2、根据权利要求1所述的核间消息通信方法，其特征在于，所述步骤A还包括：

A1、在各内核线程上建立消息类型、同步集合或异步集合以及各内核线程之间同步/异步关系的映射表；

A2、依照所述消息类型将各内核线程划分到所述同步集合或异步集合中。

3、根据权利要求2所述的核间消息通信方法，其特征在于，所述步骤A1中的消息类型单独建立在各内核线程上。

4、根据权利要求1所述的核间消息通信方法，其特征在于，所述步骤A还包括：

A3、初始化内核线程的中断分配表。

5、根据权利要求2所述的核间消息通信方法，其特征在于，所述步骤A1中的映射表内，建立各内核线程之间的关系还关联目的内核线程的状态。

6、根据权利要求3所述的核间消息通信方法，其特征在于，所述步骤

A2 中依照消息类型将各内核线程唯一的划分到所述同步集合或异步集合中。

7、根据权利要求 4 所述的核间消息通信方法，其特征在于，所述步骤 A3 中初始化过程还包括：设置处理器间中断向量和中断的优先级，并挂接各内核线程的中断服务程序。

8、根据权利要求 2 所述的核间消息通信方法，其特征在于，所述步骤 A1 中还包括：

A11、异步消息队列和同步消息缓存建立于共享内存的分配内存；

A12、异步消息队列和同步消息缓存为双向状态；

A13、异步消息队列和同步消息缓存的个数由内核线程个数  $N$  决定：

$$\text{异步消息队列个数} = N \times (N - 1)$$

$$\text{同步消息缓存个数} = N \times (N - 1) / 2。$$

9、根据权利要求 1 所述的核间消息通信方法，其特征在于，所述步骤 B 中还包括：

B1、同步消息通信方法：源内核线程封装消息，置位标志，原子操作读循环等待目的内核线程清除所述标志；

B2、异步可靠消息通信方法：源内核线程封装消息，置位标志，原子操作读循环等待目的内核线程清除所述标志；

B3、异步普通消息通信方法：源内核线程封装消息。

10、根据权利要求 1 所述的核间消息通信方法，其特征在于，所述步骤 C 中还包括：

C1、同步消息通信方法：所述处理器间中断触发目的内核线程的中断，目的内核线程解析消息，执行完成操作后通过原子操作清除所述标志；

C2、异步可靠消息通信方法：所述处理器间中断触发目的内核线程的

---

中断，目的内核线程解析消息，通过原子操作清除所述标志后再执行完成操作；

C3、异步普通消息通信方法：目的内核线程解析消息，执行完成操作。

## 一种在多核多线程处理器中的核间消息通信方法

### 技术领域

本发明涉及多核多线程处理器领域，尤其涉及一种利用多核多线程处理器的处理器间中断进行内核线程间同步和异步消息通信方法的改进。

### 背景技术

随着多核处理器被逐渐广泛的应用到更多的技术领域，其强大的并行计算能力、低功耗以及高集成度的优点也开始逐渐被人们所认知；尤其在几个主要的多核处理器厂商生产的产品系列中，双核、四核、八核、十六核已经成为各种应用开发平台的主流；甚至有的多核处理器在一个处理器内核中还包含了多个内核硬线程，能够独立运行操作系统实例，给应用开发带来了极大的灵活性，但同时也带来了一系列问题，例如在充分发挥多核处理器并行计算优势的条件下，如何保持核间或线程间的互斥和同步关系，如何保证核间或者线程间通信的及时性和可靠性等问题。

上述多核处理器中的内核根据产品的不同可能仅存在一个内核硬线程或者包含有多个内核硬线程，为保持本文的一致性，下文中都用内核线程来说明。

多核处理器的应用开发模式主要有以下三种：

1.非对称多处理模式 AMP(Asymmetric multiprocessing)，即每个内核线程运行一个独立的操作系统，根据所运行的操作系统是否相同还可细分为同构 AMP 和异构 AMP；

2.对称多处理模式 SMP(Symmetric multiprocessing)，即所有内核线程只运行在一个操作系统管理之下，可以在不同的内核线程上调度任务；

3.混合多处理模式 BMP(Bound multiprocessing), 即所有内核线程只运行在一个操作系统管理之下, 可以根据应用的需要, 指定到某个固定的内核线程运行任务, 避免由于任务在不同内核线程间的切换而带来的性能下降;

在多核处理器中, 操作系统需要在多个内核线程之间协调操作, 芯片内部一个内核线程常常要向系统中的其他内核线程发出中断请求, 这通常是靠处理器间中断 IPI(Interprocessor interrupt)实现的, 即处理器间中断可以被用来进行内核线程间的同步或异步消息通信。例如当检测到目的内核线程的就绪队列中有更高优先等级的任务等待调度时, 当前的源内核线程就发送处理器间中断来指示目标内核线程, 可能会需要一次任务的调度, 目的内核线程收到中断后, 还需要根据自身的情况作进一步判定, 是否进行所述任务的切换。这说明在该消息的处理中, 源内核线程消息发送和目的内核线程进行任务调度操作是异步关系。

再者, 当某个内存页面映射的属性发生改变时, 例如内存页面属性由可写变为只读时, 就需要更新其它内核线程的旁路转换缓冲映射 TLB(Translation Lookaside Buffer), 必须向所有的内核线程发送处理器间中断来刷新各内核线程中 TLB 内容, 此时, 源内核线程消息的发送和目的内核线程刷新 TLB 的处理必须是同步关系; 否则, 就会造成各内核线程对内存访问属性不一致, 从而可能会导致访问内存数据的错误。

在现有技术的多核处理器间的消息通信方法中, 例如中国专利文件 CN200510087321 所述的一种嵌入式实时操作系统中多核处理器的核间通信方法, 表述了源内核线程将待发送消息封装后, 放入内存的某个位置, 然后触发目的内核线程虚中断, 通过管道通知目的内核线程有消息到达; 目的内核线程从管道中解析出待处理消息的位置后, 对消息进行解包处理, 并进行下一步的操作。此时内核线程间的消息通信方式是异步关系, 而这种核间通信方法只能适用于不同内核线程间运行的操作系统任务或者内核

线程间的异步消息通信过程。

通过上述的举例说明，只有在多核处理器中根据一些关键的操作应用类型，同时选择内核线程间同步或异步的消息通信，才能更好地提高多核处理器程序应用的实时性、稳定性和可靠性，特别当在某种操作下不同内核线程间既存在同步也存在异步关系时，或者内核线程处于不同的状态下接收到相同的消息时，依靠现有的内核线程间消息通信方法就会有较为明显的局限性。因为在现有的多核多线程处理器系统消息通信方式中，还尚未考虑源内核线程与目的内核线程之间的消息发送和执行操作的同步关系，以及各内核线程之间在消息触发下的相对关系。可见，在一些关键的应用开发平台中，内核线程间这种单一的异步执行操作往往无法满足需要，而且由于操作的不同步还可能会带来严重的后果。

因此，现有技术尚有待改进和发展。

## 发明内容

本发明的目的在于提供一种在多核处理器中利用处理器间中断IPI(Interprocessor interrupt)进行多核处理器不同内核线程间的消息通信方法，通过对处理器间中断的功能进行封装，比如把中断封装成函数以实现功能，根据消息类型来划分内核线程层次关系并建立映射表，使内核线程间可根据不同的消息类型保持同步或异步执行操作，以提高系统的实时性、稳定性和可靠性。

本发明的技术方案如下：

一种在多核多线程处理器中的核间消息通信方法，其包括以下步骤：

A、确立各内核线程在各种消息类型触发下的同步或异步关系，建立各内核线程之间异步消息队列和同步消息缓存；

B、源内核线程读取当前消息所对应的内核线程同步集合或异步集合，将封装的消息体放入所述异步消息队列和同步消息缓存中；

C、目的内核线程收到处理器间中断后，从所述异步消息队列和同步消息缓存中解析消息内容并执行完成操作。

所述的核间消息通信方法，其中，所述步骤 A 还包括：

A1、在各内核线程上建立消息类型、同步集合或异步集合以及各内核线程之间同步/异步关系的映射表；

A2、依照所述消息类型将各内核线程划分到所述同步集合或异步集合中。

所述的核间消息通信方法，其中，所述步骤 A1 中的消息类型单独建立在各内核线程上。

所述的核间消息通信方法，其中，所述步骤 A 还包括：

A3、初始化内核线程的中断分配表。

所述的核间消息通信方法，其中，所述步骤 A1 中的映射表内，建立各内核线程之间的关系还关联目的内核线程的状态。

所述的核间消息通信方法，其中，所述步骤 A2 中依照消息类型将各内核线程唯一的划分到所述同步集合或异步集合中。

所述的核间消息通信方法，其中，所述步骤 A3 中初始化过程还包括：设置处理器间中断向量和中断的优先级，并挂接各内核线程的中断服务程序。

所述的核间消息通信方法，其中，所述步骤 A1 中还包括：

A11、异步消息队列和同步消息缓存建立于共享内存的分配内存；

A12、异步消息队列和同步消息缓存为双向状态；

A13、异步消息队列和同步消息缓存的个数由内核线程个数 N 决定：

异步消息队列个数= $N \times (N - 1)$

同步消息缓存个数= $N \times (N - 1) / 2$ 。

所述的核间消息通信方法，其中，所述步骤 B 中还包括：

B1、同步消息通信方法：源内核线程封装消息，置位标志，原子操作



读循环等待目的内核线程清除所述标志;

B2、异步可靠消息通信方法: 源内核线程封装消息, 置位标志, 原子操作读循环等待目的内核线程清除所述标志;

B3、异步普通消息通信方法: 源内核线程封装消息。

所述的核间消息通信方法, 其中, 所述步骤 C 中还包括:

C1、同步消息通信方法: 所述处理器间中断触发目的内核线程的中断, 目的内核线程解析消息, 执行完成操作后通过原子操作清除所述标志;

C2、异步可靠消息通信方法: 所述处理器间中断触发目的内核线程的中断, 目的内核线程解析消息, 通过原子操作清除所述标志后再执行完成操作;

C3、异步普通消息通信方法: 目的内核线程解析消息, 执行完成操作。

本发明所提供的一种在多核多线程处理器中的核间消息通信方法, 由于采用了多核多线程处理器的处理器间中断封装的同步/异步消息通信模式, 突破了现有的内核线程间消息通信方法的局限性, 考虑到了源内核线程与目的内核线程之间的消息发送和执行操作的同步关系, 保证了各内核线程可在不同应用场景中操作执行正确的相对关系, 满足了某些关键的应用平台的开发需要, 极大提高了多核多线程系统运行的可靠性和灵活性。

## 附图说明

图 1 是本发明多核处理器内核线程的一种应用模型示意图;

图 2 是本发明在消息触发下内核线程间的同步/异步集合划分示意图;

图 3 是本发明在消息触发下内核线程间的一种层次关系示意图;

图 4 是本发明在消息触发下内核线程间的另一种层次关系示意图;

图 5 是本发明在消息触发下内核线程同步/异步集合映射表示意图;

图 6 是本发明多核处理器内核线程消息通信模型示意图;

图 7 是本发明内核线程发送同步/异步消息通信软件流程图;

图 8 是本发明内核线程接收同步/异步消息通信软件流程图。

### 具体实施方式

以下将结合附图所示，对本发明多核多线程处理器中的核间消息通信方法加以详细说明。

本发明多核多线程处理器中核间消息通信方法的具体实施方式，主要改进在，利用处理器间中断 IPI(Interprocessor interrupt)实现源内核线程与目的内核线程之间的消息发送和执行操作的异步/同步关系，至于多核多线程处理器的制作和应用过程为现有技术所熟知，在此不再赘述。

本发明多核多线程处理器中的核间消息通信方法的具体实施方式中，多核多线程处理器的多个内核线程在不同的应用场景中具有很大的灵活性，在交换机或路由器上的典型功能划分如图 1 所示，内核线程 200 既可以作为控制内核线程，也可以作为报文分类内核线程或发送调度内核线程或报文转发内核线程，具体内核线程的功能类型和数量可根据实际要求进行配置。而图 2 所示，表明在不同消息类型的触发下，所述内核线程可以唯一的划分到某个内核线程的同步集合或者异步集合中。

而对每个内核线程，在消息触发下对应的同步集合和异步集合也不一定相同，所以首先必须确立各内核线程在各种消息类型触发下的同步或异步关系，在各内核线程上建立单独的消息类型与每一个内核线程之间同步/异步关系的映射表，依照消息的类型将各内核线程唯一地划分到每个内核线程上建立的同步集合或异步集合中。

如图 3 和图 4 所示，为两种消息触发下各内核线程间的层次关系，由映射表来决定每个内核线程间在该消息触发下的同步或异步关系，例如内核线程 300 与内核线程 400 是同步关系，内核线程 400 与内核线程 500 是异步关系，但是内核线程 300 与内核线程 500 无直接的同步/异步关系。

图 5 所示的映射表，进一步说明了依照所述消息类型的不同，将不同状

态下的内核线程唯一的划分到所述同步集合或异步集合中，例如状态 1 下消息类型 1 触发的同步内核线程 4 唯一的被划分到内核线程 0 的同步集合中。较好的是，所述映射表内，在建立各内核线程之间的关系时兼顾考虑到目的内核线程的状态。

如图 6 所示的内核线程间消息通讯模型，内核线程间在共享内存的分配内存中建立双向的异步消息队列和同步消息缓冲区，源内核线程根据映射表读取当前消息对应的内核线程同步集合和异步集合，使用同步消息通信的方法发送消息到同步集合中包含的内核线程，将封装后的消息体放入同步缓冲区中；使用合适的异步消息通信方法发送消息到异步集合中包含的内核线程，将封装后的消息体放入异步消息队列中；目的内核线程收到处理器间中断后，从异步消息队列或者同步缓冲区中获取所述消息体的位置和具体内容，然后根据当前接收到消息查找当前内核线程的映射表，再触发下一步的消息发送。其完整的步骤过程如下：

第 1 步，由于多核处理器有多个内核线程，而不同内核线程在各种应用模式下所处的位置也不尽相同，例如，在 AMP(Asymmetric multiprocessing) 模式下既有作为控制面内核线程，也有作为数据面的转发内核线程，还有作为报文保序、QOS(Quality of Service)功能的的内核线程，内核线程之间的关系按照处理顺序既有并行的也有串行的，可以说，各个内核线程之间的关系依照应用模式的不同而有所不同。必须确立各内核线程在各种消息类型触发下的同步或异步关系，以及在各内核线程上建立单独的消息类型和每一个内核线程之间同步/异步关系的映射表；并依照所述消息的类型将各内核线程唯一地划分到每个内核线程上建立的同步集合或异步集合中；

第 2 步，映射表中内核线程间的关系还需考虑目的内核线程的状态，因为目的内核线程虽然接收到同样的消息类型和消息内容，但在内核线程正常运行状态和内核线程异常状态所进行的操作会有所不同；

第 3 步，实现内核线程间同步消息通信的方法：源内核线程封装消息

后，置位标志，通过处理器间中断触发目的内核线程的中断，源内核线程原子操作读循环等待目的内核线程清除标志；目的内核线程解析消息，执行完成操作后，通过原子操作清除标志。为防止死循环，可在源内核线程等待中增加等待超时，当等待时间超出后，源内核线程退出等待；

第 4 步，实现内核线程间异步可靠消息通信的方法：源内核线程封装消息后，置位标志，通过处理器间中断触发目的内核线程的中断，源内核线程原子操作读循环等待目的内核线程清除标志；目的内核线程解析消息，通过原子操作清除标志后再执行完成操作。为防止死循环，可在源内核线程等待中增加等待超时，当等待时间超出后，源内核线程退出等待；

第 5 步，实现内核线程间异步普通消息通信的方法：源内核线程封装消息后，通过处理器间中断触发目的内核线程的中断，目的内核线程解析消息，然后执行完成操作；

第 6 步，正确初始化各内核线程的中断分配表，包括设置处理器间中断向量和中断的优先级，并挂接各内核线程的中断服务程序；

第 7 步，在共享内存中分配内存，用于建立各内核线程间的双向异步消息队列和同步消息缓存；假定内核线程个数为  $N$ ，异步消息队列和同步缓冲区的个数分别为  $N * (N-1)$  和  $N * (N-1) / 2$ ；

第 8 步，源内核线程根据映射表读取当前消息对应的内核线程同步集合和异步集合，使用同步消息通信的方法发送消息到同步集合中包含的内核线程，将封装的消息体放入同步缓冲区中；使用适当的异步消息通信方法发送消息到异步集合中包含的内核线程，将封装的消息体放入异步消息队列下；目的内核线程收到处理器间中断后，从所述异步消息队列或者同步缓冲区中获取消息具体内容，然后根据当前收到消息对应的当前内核线程同步集合或异步集合，再触发下一步的消息发送。

结合发送和接收两种应用，将前述步骤的软件流程说明如下：

图 7 所示，本发明多核多线程处理器中核间消息通信方法的具体实施

方式所述内核线程间消息发送的软件流程:

步骤 S11: 进入内核线程间消息发送的软件流程, 源内核线程准备发送消息;

步骤 S12: 根据内核线程当前的状态和待发送消息的类型检索当前消息的映射表, 查找同步集合或异步集合;

步骤 S13: 判断待发送消息的同步集合是否为空, 是则进入步骤 S16, 否则下一步;

步骤 S14: 如待发送消息的同步集合不为空, 则向目的内核线程发送同步消息, 包括封装消息体到同步缓存中, 置位等待标志, 触发同步集合中所有目的内核线程的处理器间中断, 循环等待所有目的内核线程的应答;

步骤 S15: 判断目的内核线程同步执行是否完毕或等待应答是否超时, 是则下一步, 否则返回上一步;

步骤 S16: 判断待发送消息的异步集合是否为空, 是则进入步骤 S19, 否则下一步;

步骤 S17: 如待发送消息的异步集合不为空, 则向目的内核线程发送异步消息, 包括封装消息体到异步消息队列中, 根据消息的可靠程度决定是否置位等待标志, 触发异步集合中所有目的内核线程的处理器间中断, 如是异步可靠消息则循环等待所有目的内核线程的应答;

步骤 S18: 判断目的内核线程是否需等待应答或者等待应答是否超时, 是则下一步, 否则返回上一步;

步骤 S19: 退出内核线程间消息发送的软件流程。

图 8 所示, 本发明多核多线程处理器中核间消息通信方法的具体实施方式所述内核线程间消息接收的软件流程:

步骤 S21: 进入内核线程间消息接收的软件流程; 内核线程接收到处理器间中断;

步骤 S22: 转入中断服务程序, 包括设置处理器间中断向量和中断的优

优先级，并挂接各内核线程的中断服务程序；

步骤 S23a: 判断是否收到同步消息，是则进入步骤 S24a，否则下一步；

步骤 S23b: 判断是否收到异步消息，是则进入步骤 S24b，否则进入步骤 S212；

步骤 S24a: 从同步消息缓冲区中获取当前消息，根据内核线程当前的状态和消息类型检索当前消息的映射表，获取当前消息对应的同步集合；

步骤 S24b: 从异步消息队列中获取当前消息，根据内核线程当前的状态和消息类型检索当前消息的映射表，获取当前消息对应的异步集合；

步骤 S25: 判断所接收消息的同步集合是否为空，是则进入步骤 S28，否则下一步；

步骤 S26: 如所接收消息的同步集合不为空，则向目的内核线程发送同步消息，包括封装消息体到同步缓存中，置位等待标志，触发同步集合中所有目的内核线程的处理器间中断，循环等待所有目的内核线程的应答；

步骤 S27: 判断目的内核线程同步执行是否完毕或等待应答是否超时，是则下一步，否则返回上一步；

步骤 S28: 接收同步消息需应答源内核线程，清除标志；接收异步消息需根据可靠类型给予应答；

步骤 S29: 判断所接收消息的异步集合是否为空，是则进入步骤 S212，否则下一步；

步骤 S210: 如所接收消息的异步集合不为空，则向目的内核线程发送异步消息，包括封装消息体到异步消息队列中，根据消息的可靠程度决定是否置位等待标志，触发异步集合中所有目的内核线程的处理器间中断，如是异步可靠消息则循环等待所有目的内核线程的应答；

步骤 S211: 判断目的内核线程是否需等待应答或者等待应答是否超时，是则下一步，否则返回上一步；

步骤 S212: 退出内核线程间消息接收的软件流程。

本发明具体实施方式所描述多核多线程处理器中的核间消息通信方法，根据实际应用还可以采用现有各种可能的方案，其基本手段为本领域技术人员所熟知，在此不再赘述。

本发明多核多线程处理器中核间消息通信方法的具体实施方式中，由于采用了多核多线程处理器的处理器间中断封装的同步/异步消息通信模式，突破了现有的内核线程间消息通信方法的局限性，考虑到了源内核线程与目的内核线程之间的消息发送和执行操作的同步关系，保证了各内核线程可在不同应用场景中操作执行正确的相对关系，满足了某些关键的应用平台的开发需要，极大提高了多核多线程系统运行的可靠性和灵活性。

应当理解的是，对本领域普通技术人员来说，可以根据上述方案的说明加以改进或变换，例如建立各内核线程间的双向异步消息队列和同步消息缓存的个数变化，以及先异步消息通讯再同步消息通讯等，而所有这些改进和变换都本应属于本发明所附权利要求的保护范围。

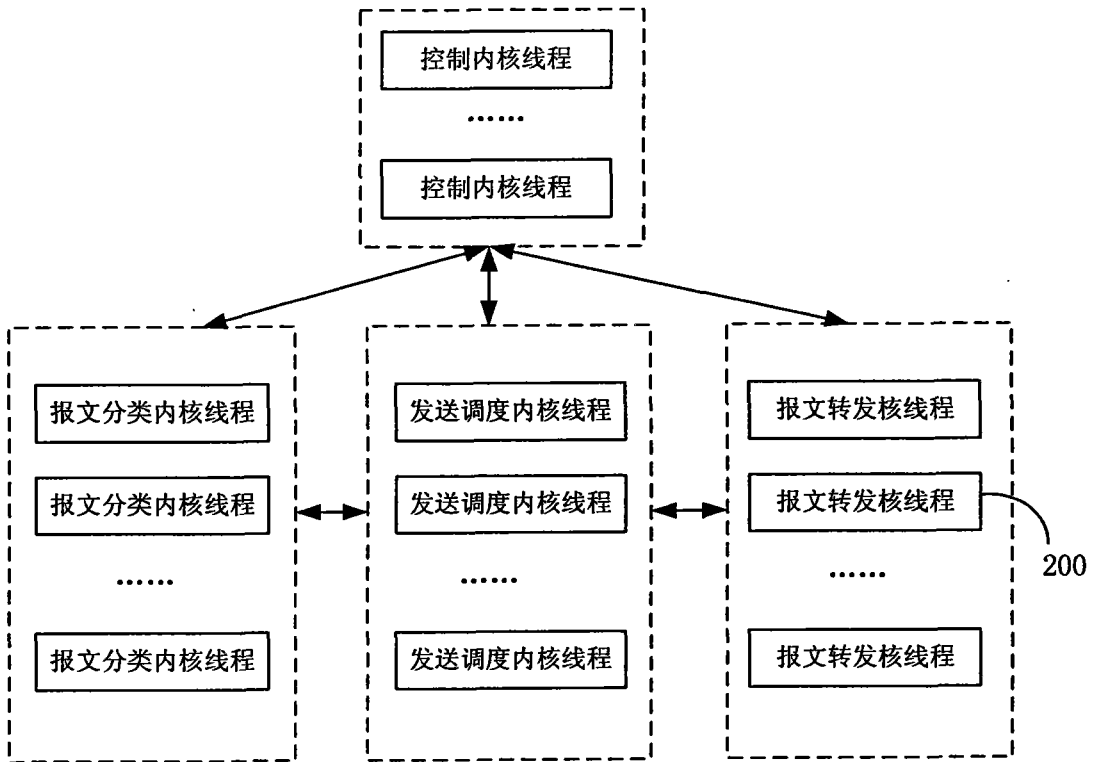


图 1

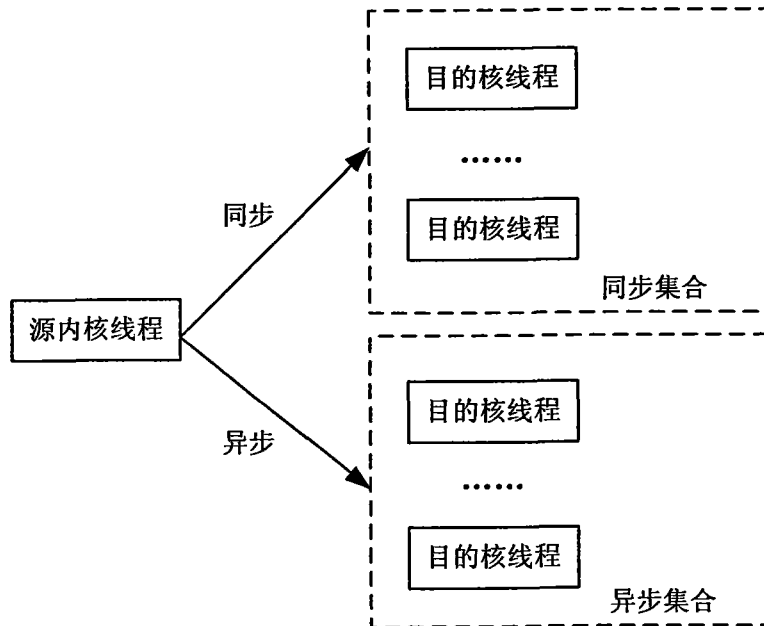


图 2



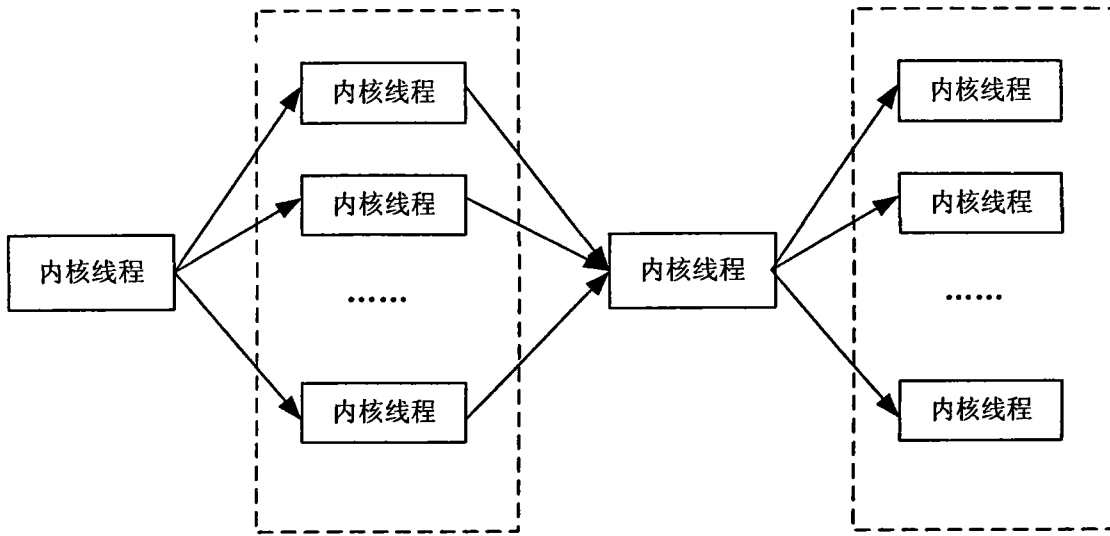


图 3

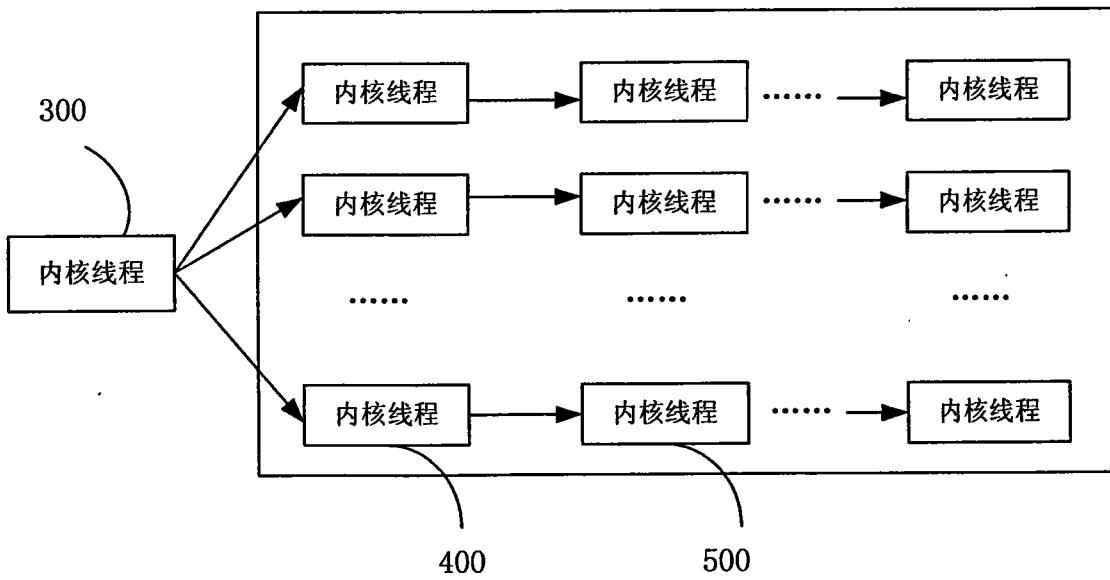


图 4

内核线程0的映射表

	消息类型	同步集合	异步集合
状态1	类型0	内核线程1、2、5	内核线程3、7
	类型1	内核线程3、4	内核线程2、6
	类型2	无	内核线程1、2、3、4、5、6、7
	.....	.....	.....
	类型N	内核线程7	内核线程1、3、4、5
状态2	类型0	内核线程1、2、3、4、5、6、7	无
	类型1	无	无
	类型2	无	无
	.....	.....	.....
状态M	类型0	无	内核线程4、6
	类型1	无	内核线程1、2
	类型2	无	内核线程1、2、3、4、5、6、7
	.....	.....	.....
	类型N	无	内核线程4、5

图 5

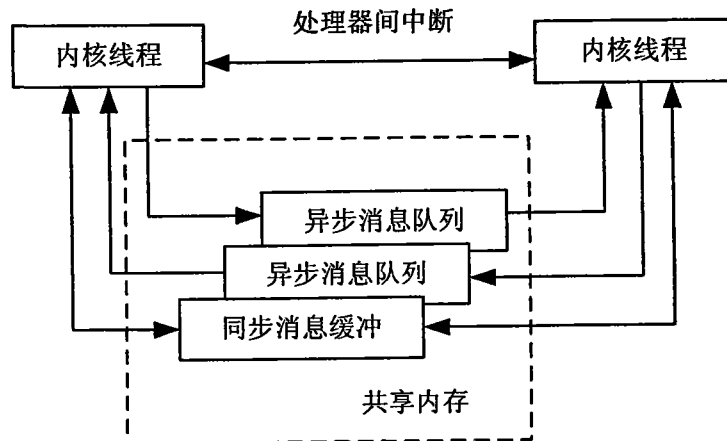


图 6

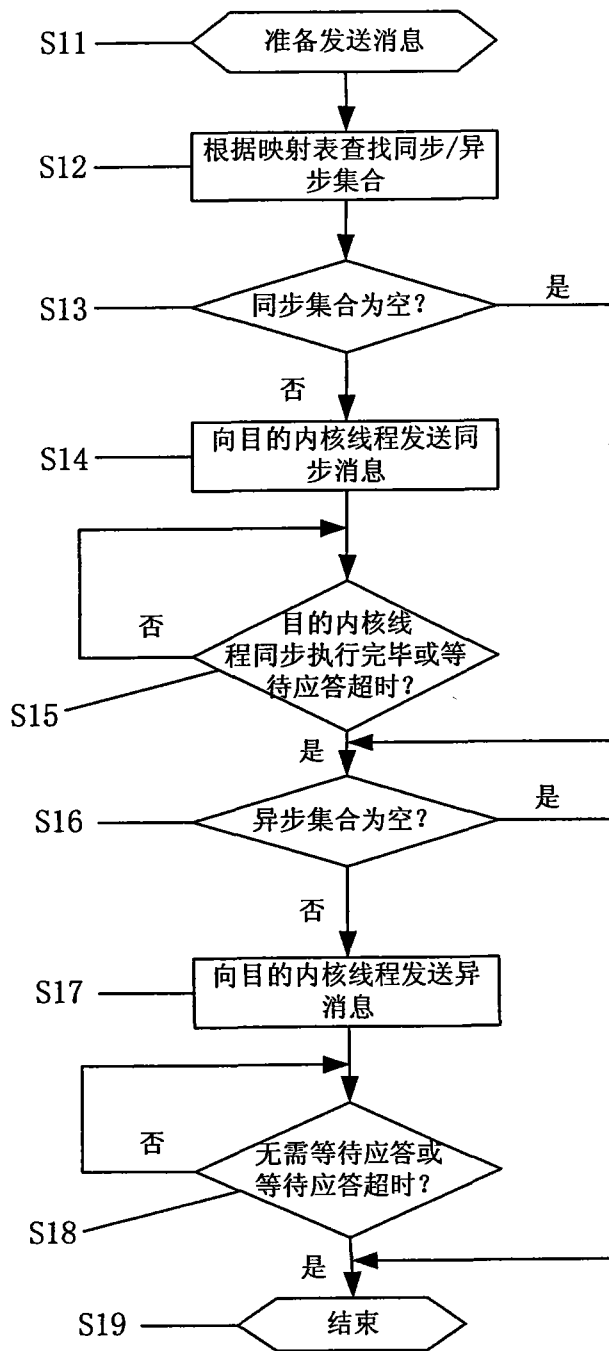


图 7

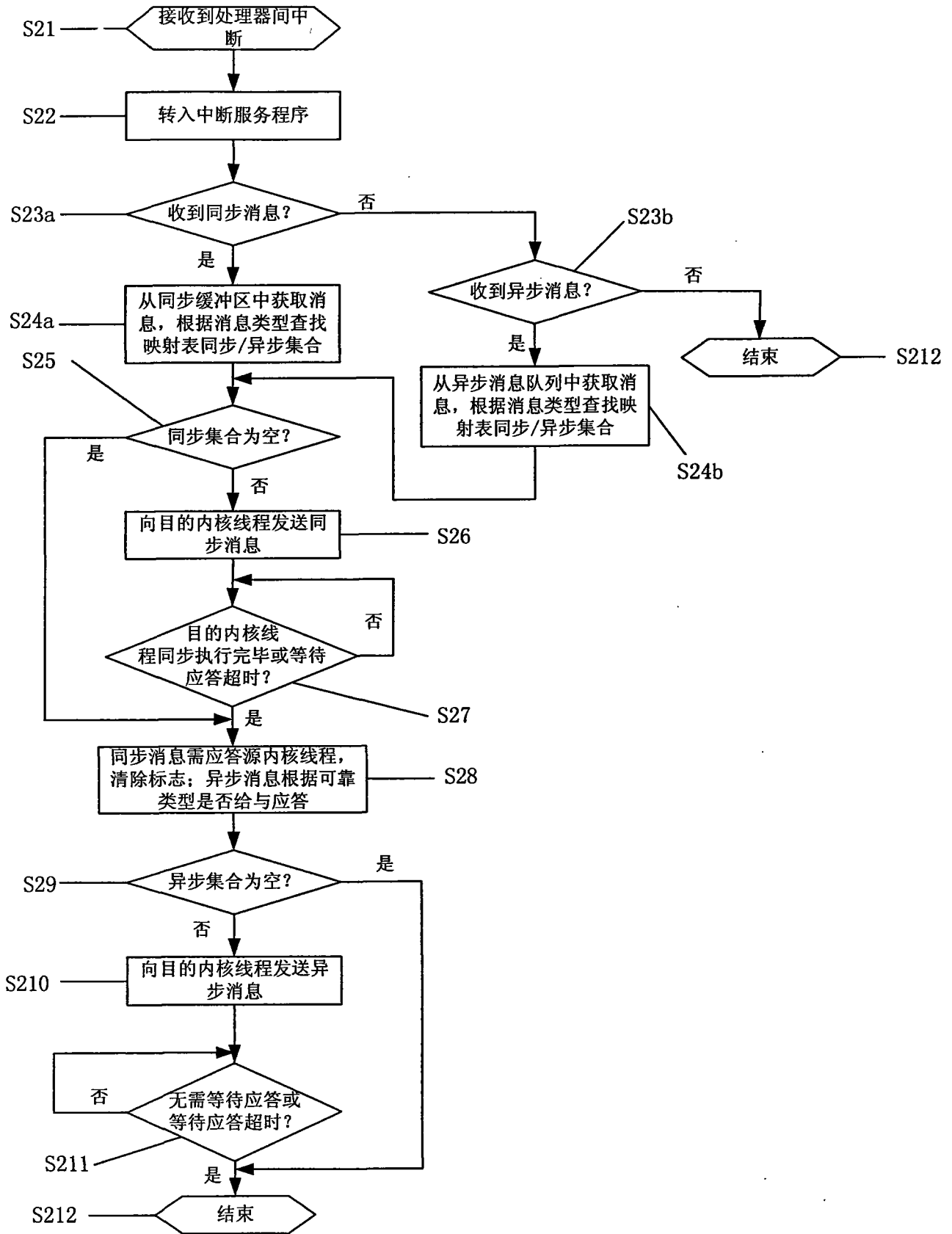


图 8