



(12) 发明专利申请

(10) 申请公布号 CN 118349283 A

(43) 申请公布日 2024.07.16

(21) 申请号 202410469221.7

(22) 申请日 2024.04.18

(71) 申请人 清华大学

地址 100084 北京市海淀区清华园

(72) 发明人 刘学 施路平 赵蓉 李晓民

(74) 专利代理机构 北京三友知识产权代理有限公司 11127

专利代理师 张瑜

(51) Int. Cl.

G06F 9/38 (2018.01)

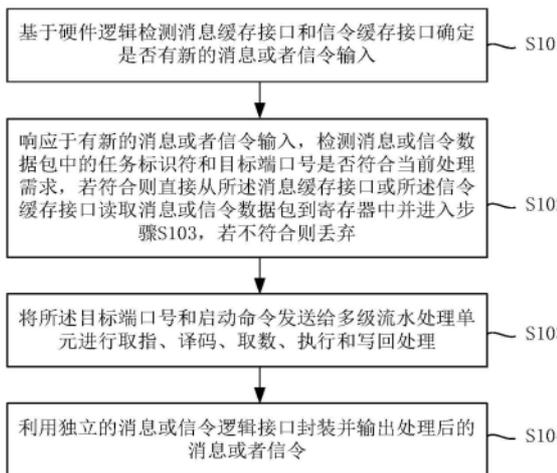
权利要求书3页 说明书13页 附图4页

(54) 发明名称

用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法和装置

(57) 摘要

本发明提供了一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法和装置,方法包括:检测消息缓存接口和信令缓存接口确定是否有新的消息或信令输入;响应于有新的消息或信令输入,检测消息或信令数据包中的任务标识符和目标端口号是否符合当前处理需求,若符合则直接从所述消息缓存接口或所述信令缓存接口读取消息或信令数据包到寄存器中并进入步骤S103,若不符合则丢弃;将所述目标端口号和启动命令发送给多级流水处理单元进行取指、译码、取数、执行和写回处理;输出处理后的消息或者信令。本发明将计算指令和通信指令进行了解耦,利用消息缓存接口和信令缓存接口解耦接收并处理计算指令和通信指令,基于控制逻辑单元和多级流水处理单元完成指令的并行处理,可以实现网络系统的通信效率提升,提高分布式集群系统处理并行任务效率。



1. 一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,其特征在于,所述方法包括:

基于硬件逻辑检测消息缓存接口和信令缓存接口确定是否有新的消息或者信令输入;

响应于有新的消息或者信令输入,检测消息或信令数据包中的任务标识符和目标端口号是否符合当前处理需求,若符合则直接从所述消息缓存接口或所述信令缓存接口读取消息或信令数据包到寄存器中并进入下一阶段处理,若不符合则丢弃;

将所述目标端口号和启动命令发送给多级流水处理单元进行取指、译码、取数、执行和写回处理;

利用独立的消息或信令逻辑接口封装并输出处理后的消息或者信令。

2. 如权利要求1所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,其特征在于,所述方法还包括:若检测到新输入的消息为触发消息,则根据消息内容和预先配置的偏移量,定位到对应的子程序区进行处理。

3. 如权利要求1所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,其特征在于,所述方法还包括:当所述消息缓存接口和信令缓存接口为空的时候,在可用的端口号所对应的宏指令程序之间进行仲裁,确定一个端口号后将确定的端口号及启动命令发送给多级流水处理单元。

4. 如权利要求1所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,其特征在于,所述多级流水处理单元包括取指模块、译码模块、取数模块、执行模块和写回模块,所述的将所述目标端口号和启动命令发送给多级流水处理单元进行取指、译码、取数、执行和写回处理包括:

将所述目标端口号和启动命令发送给取指模块,由所述取指模块根据所述目标端口号查询对应的指令地址寄存器以获得指令,并将所述指令发送给译码模块;

所述译码模块接收到指令后进行初步译码得到指令码,并将所述指令码输出给取数模块,若所述译码模块遇到流水线停顿的状态,则进入延迟处理状态,并在下一个周期再输出译码结果;

所述取数模块根据指令码解析出执行模块和写回模块两级逻辑需要的操作码并且向后传输;

所述执行模块基于所述操作码完成指令的执行;

所述写回模块基于所述执行模块的执行结果和所述操作码完成写入操作、栈操作以及控制流操作。

5. 如权利要求4所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,其特征在于,所述方法还包括:

当所述取指模块检测到指令访存冲突的时候停止本级功能的执行,待冲突解决后再恢复本级功能的执行。

6. 如权利要求5所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,其特征在于,当所述取指模块检测到指令访存冲突的时候,所述方法还包括:将访存冲突的指令进行暂时压栈处理,直到冲突解决后从栈中恢复所述访存冲突的指令并继续执行。

7. 如权利要求1所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行

方法,其特征在于,所述方法还包括:

根据所述消息或信令的当前执行情况和资源占用状态动态分配所述非阻塞型宏指令多级流水处理器的资源。

8.如权利要求4所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,其特征在于,所述取数模块还用于接收所述写回模块产生的流水线延迟、流水线丢弃以及原子操作结束信号;

当收到原子操作结束信号时,所述取数模块丢弃当前操作,并交还控制权给所述非阻塞型宏指令多级流水处理器;

当收到流水线丢弃信号时,所述取数模块丢弃当前操作,并继续执行后续指令;

所述取指模块和所述译码模块还用于接收所述写回模块产生的原子操作结束信号,并基于所述原子操作结束信号丢弃当前操作,然后交还控制权给所述非阻塞型宏指令多级流水处理器。

9.如权利要求8所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,其特征在于,当所述写回模块产生原子操作结束信号后,所述非阻塞型宏指令多级流水处理器将当前状态进行暂存,接着接收并解析新的消息或信令输入,基于解析的结果由多级流水处理单元进行新一轮流水操作。

10.一种用于分布式集群系统的非阻塞型宏指令多级流水处理器,其特征在于,所述非阻塞型宏指令多级流水处理器包括:

控制逻辑单元,用于基于硬件逻辑检测消息缓存接口和信令缓存接口确定是否有新的消息或者信令输入;响应于有新的消息或者信令输入,检测消息或信令数据包中的任务标识符和目标端口号是否符合当前处理需求,若符合则直接从所述消息缓存接口或所述信令缓存接口读取消息或信令数据包到寄存器中并将所述目标端口号和启动命令发送给多级流水处理单元,若不符合则丢弃;

多级流水处理单元,用于基于所述目标端口号和启动命令进行取指、译码、取数、执行和写回处理;

输出单元,用于利用独立的消息或信令逻辑接口封装并输出处理后的消息或者信令。

11.如权利要求10所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器,其特征在于,所述控制逻辑单元还用于:若检测到新输入的消息为触发消息,则根据消息内容和预先配置的偏移量,定位到对应的子程序区进行处理。

12.如权利要求10所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器,其特征在于,所述控制逻辑单元还用于:当所述消息缓存接口和信令缓存接口为空的时候,在可用的端口号所对应的宏指令程序之间进行仲裁,确定一个端口号后将确定的端口号及启动命令发送给所述多级流水处理单元。

13.如权利要求10所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器,其特征在于,所述多级流水处理单元包括取指模块、译码模块、取数模块、执行模块和写回模块,

所述取指模块用于根据所述目标端口号查询对应的指令地址寄存器以获得指令,并将所述指令发送给译码模块;

所述译码模块用于接收到指令后进行初步译码得到指令码,并将所述指令码输出给取

数模块,若所述译码模块遇到流水线停顿的状态,则进入延迟处理状态,并在下一个周期再输出译码结果;

所述取数模块用于根据指令码解析出执行模块和写回模块两级逻辑需要的操作码并且向后传输;

所述执行模块用于基于所述操作码完成指令的执行;

所述写回模块用于基于所述执行模块的执行结果和所述操作码完成写入操作、栈操作以及控制流操作。

14. 如权利要求13所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器,其特征在于,所述取指模块还用于检测到指令访存冲突的时候停止本级功能的执行,待冲突解决后再恢复本级功能的执行。

15. 如权利要求14所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器,其特征在于,当所述取指模块检测到指令访存冲突的时候,所述装置还包括压栈处理单元:用于将访存冲突的指令进行暂时压栈处理,直到冲突解决后从栈中恢复所述访存冲突的指令并继续执行。

16. 如权利要求10所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器,其特征在于,所述装置还包括资源调度单元:用于根据所述消息或信令的当前执行情况和资源占用状态动态分配所述非阻塞型宏指令多级流水处理器的资源。

17. 如权利要求13所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器,其特征在于,所述取数模块还用于接收所述写回模块产生的流水线延迟、流水线丢弃以及原子操作结束信号;

当收到原子操作结束信号时,所述取数模块丢弃当前操作,并交还控制权给所述非阻塞型宏指令多级流水处理器;

当收到流水线丢弃信号时,所述取数模块丢弃当前操作,并继续执行后续指令;

所述取指模块和所述译码模块还用于接收所述写回模块产生的原子操作结束信号,并基于所述原子操作结束信号丢弃当前操作,然后交还控制权给所述非阻塞型宏指令多级流水处理器。

18. 如权利要求17所述的用于分布式集群系统的非阻塞型宏指令多级流水处理器,其特征在于,当所述写回模块产生原子操作结束信号后,所述非阻塞型宏指令多级流水处理器将当前状态进行暂存,接着接收并解析新的消息或信令输入,基于解析的结果由多级流水处理单元进行新一轮流水操作。

19. 一种电子设备,包括存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1至9任一项所述方法的步骤。

20. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1至9任一项所述方法的步骤。

用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法和装置

技术领域

[0001] 本发明涉及宏指令处理器技术领域,尤其涉及一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法和装置。

背景技术

[0002] 当前分布式深度学习平台的通信计算模式较为单一,通信指令和计算指令不能有效结合,造成了通信和计算的交互延迟。这里的通信指令是指涉及数据在分布式系统的各个节点之间的传输的各种指令,例如,同步信令、生产消费型通信类型、参数同步、梯度更新消息的发送与接收等。而计算指令是指包括执行具体的数学运算和模型训练步骤的指令,比如前向传播、反向传播等计算任务。这里的通信指令和计算指令不能有效结合,是指通信指令和计算指令高度耦合,而这意味着将影响计算和通信操作的并行性和效率,进而导致了通信和计算的交互延迟。另外目前分布式集群系统经常会出现阻塞的情况,如果某个节点因为阻塞而无法处理新的请求,那么该节点资源可能会被浪费。

发明内容

[0003] 有鉴于此,本发明提供一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法和装置,以解决上述提及的至少一个问题。

[0004] 为了实现上述目的,本发明采用以下方案:

[0005] 根据本发明的第一方面,提供一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,所述方法包括:基于硬件逻辑检测消息缓存接口和信令缓存接口确定是否有新的消息或者信令输入;响应于有新的消息或者信令输入,检测消息或信令数据包中的任务标识符和目标端口号是否符合当前处理需求,若符合则直接从所述消息缓存接口或所述信令缓存接口读取消息或信令数据包到寄存器中并进入下一阶段处理,若不符合则丢弃;将所述目标端口号和启动命令发送给多级流水处理单元进行取指、译码、取数、执行和写回处理;利用独立的消息或信令逻辑接口封装并输出处理后的消息或者信令。

[0006] 作为本发明的一个实施例,上述方法还包括:若检测到新输入的消息为触发消息,则根据消息内容和预先配置的偏移量,定位到对应的子程序区进行处理。

[0007] 作为本发明的一个实施例,上述方法还包括:当所述消息缓存接口和信令缓存接口为空的时候,在可用的端口号所对应的宏指令程序之间进行仲裁,确定一个端口号后将确定的端口号及启动命令发送给多级流水处理单元。

[0008] 作为本发明的一个实施例,上述多级流水处理单元包括取指模块、译码模块、取数模块、执行模块和写回模块,所述的将所述目标端口号和启动命令发送给多级流水处理单元进行取指、译码、取数、执行和写回处理包括:将所述目标端口号和启动命令发送给取指模块,由所述取指模块根据所述目标端口号查询对应的指令地址寄存器以获得指令,并将所述指令发送给译码模块;所述译码模块接收到指令后进行初步译码得到指令码,并将所

述指令码输出给取数模块,若所述译码模块遇到流水线停顿的状态,则进入延迟处理状态,并在下一个周期再输出译码结果;所述取数模块根据指令码解析出执行模块和写回模块两级逻辑需要的操作码并且向后传输;所述执行模块基于所述操作码完成指令的执行;所述写回模块基于所述执行模块的执行结果和所述操作码完成写入操作、栈操作以及控制流操作。

[0009] 作为本发明的一个实施例,上述方法还包括:当所述取指模块检测到指令访存冲突的时候停止本级功能的执行,待冲突解决后再恢复本级功能的执行。

[0010] 作为本发明的一个实施例,当所述取指模块检测到指令访存冲突的时候,上述方法还包括:将访存冲突的指令进行暂时压栈处理,直到冲突解决后从栈中恢复所述访存冲突的指令并继续执行。

[0011] 作为本发明的一个实施例,上述方法还包括:根据所述消息或信令的当前执行情况和资源占用状态动态分配所述非阻塞型宏指令多级流水处理器的资源。

[0012] 作为本发明的一个实施例,上述取数模块还用于接收所述写回模块产生的流水线延迟、流水线丢弃以及原子操作结束信号;当收到原子操作结束信号时,所述取数模块丢弃当前操作,并交还控制权给所述非阻塞型宏指令多级流水处理器;当收到流水线丢弃信号时,所述取数模块丢弃当前操作,并继续执行后续指令;所述取指模块和所述译码模块还用于接收所述写回模块产生的原子操作结束信号,并基于所述原子操作结束信号丢弃当前操作,然后交还控制权给所述非阻塞型宏指令多级流水处理器。

[0013] 作为本发明的一个实施例,当所述写回模块产生原子操作结束信号后,所述非阻塞型宏指令多级流水处理器将当前状态进行暂存,接着接收并解析新的消息或信令输入,基于解析的结果由多级流水处理单元进行新一轮流水操作。

[0014] 根据本发明的第二方面,提供一种用于分布式集群系统的非阻塞型宏指令多级流水处理器,所述非阻塞型宏指令多级流水处理器包括:控制逻辑单元,用于基于硬件逻辑检测消息缓存接口和信令缓存接口确定是否有新的消息或者信令输入;响应于有新的消息或者信令输入,检测消息或信令数据包中的任务标识符和目标端口号是否符合当前处理需求,若符合则直接从所述消息缓存接口或所述信令缓存接口读取消息或信令数据包到寄存器中并将所述目标端口号和启动命令发送给多级流水处理单元,若不符合则丢弃;多级流水处理单元,用于基于所述目标端口号和启动命令进行取指、译码、取数、执行和写回处理;输出单元,用于利用独立的消息或信令逻辑接口封装并输出处理后的消息或者信令。

[0015] 作为本发明的一个实施例,上述控制逻辑单元还用于:若检测到新输入的消息为触发消息,则根据消息内容和预先配置的偏移量,定位到对应的子程序区进行处理。

[0016] 作为本发明的一个实施例,上述控制逻辑单元还用于:当所述消息缓存接口和信令缓存接口为空的时候,在可用的端口号所对应的宏指令程序之间进行仲裁,确定一个端口号后将确定的端口号及启动命令发送给所述多级流水处理单元。

[0017] 作为本发明的一个实施例,上述多级流水处理单元包括取指模块、译码模块、取数模块、执行模块和写回模块,所述取指模块用于根据所述目标端口号查询对应的指令地址寄存器以获得指令,并将所述指令发送给译码模块;所述译码模块用于接收到指令后进行初步译码得到指令码,并将所述指令码输出给取数模块,若所述译码模块遇到流水线停顿的状态,则进入延迟处理(Delay)状态,并在下一个周期再输出译码结果;所述取数模块用

于根据指令码解析出执行模块和写回模块两级逻辑需要的操作码并且向后传输;所述执行模块用于基于所述操作码完成指令的执行;所述写回模块用于基于所述执行模块的执行结果和所述操作码完成写入操作、栈操作以及控制流操作。

[0018] 作为本发明的一个实施例,上述取指模块还用于检测到指令访存冲突的时候停止本级功能的执行,待冲突解决后再恢复本级功能的执行。

[0019] 作为本发明的一个实施例,当所述取指模块检测到指令访存冲突的时候,所述装置还包括压栈处理单元:用于将访存冲突的指令进行暂时压栈处理,直到冲突解决后从栈中恢复所述访存冲突的指令并继续执行。

[0020] 作为本发明的一个实施例,所述装置还包括资源调度单元:用于根据所述消息或信令的当前执行情况和资源占用状态动态分配所述非阻塞型宏指令多级流水处理器的资源。

[0021] 作为本发明的一个实施例,上述取数模块还用于接收所述写回模块产生的流水线延迟、流水线丢弃以及原子操作结束信号;当收到原子操作结束信号时,所述取数模块丢弃当前操作,并交还控制权给所述非阻塞型宏指令多级流水处理器;当收到流水线丢弃信号时,所述取数模块丢弃当前操作,并继续执行后续指令;所述取指模块和所述译码模块还用于接收所述写回模块产生的原子操作结束信号,并基于所述原子操作结束信号丢弃当前操作,然后交还控制权给所述非阻塞型宏指令多级流水处理器。

[0022] 作为本发明的一个实施例,当所述写回模块产生原子操作结束信号后,所述非阻塞型宏指令多级流水处理器将当前状态进行暂存,接着接收并解析新的消息或信令输入,基于解析的结果由多级流水处理单元进行新一轮流水操作。

[0023] 根据本发明的第三方面,提供一种电子设备,包括存储器、处理器以及存储在所述存储器上并可在所述处理器上运行的计算机程序,处理器执行所述计算机程序时实现上述方法的步骤。

[0024] 根据本发明的第四方面,提供一种计算机可读存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现上述方法的步骤。

[0025] 由上述技术方案可知,本发明提供的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法和装置,将计算指令和通信指令进行了解耦,通过数据通道及信令通道分别传输计算指令和通信指令,并利用消息缓存接口和信令缓存接口解耦接收计算指令和通信指令,基于控制逻辑单元和多级流水处理单元完成指令的并行处理,应用于分布式集群系统可以实现网络系统的通信效率提升。另外,本申请通过有效的资源调度和冲突检测机制,能够实现多个指令之间的资源争用和数据依赖关系,从而不会导致阻塞,且原子操作结束信号后,可以将当前状态进行暂存,接着接收并解析新的消息或信令输入,基于解析的结果由多级流水处理单元进行新一轮流水操作,这也使得非阻塞宏指令处理器能够有效地处理并行任务,进而提高了执行效率和响应速度。

附图说明

[0026] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以

根据这些附图获得其他的附图。在附图中：

[0027] 图1是本申请实施例提供的一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法的流程示意图；

[0028] 图2是本申请实施例提供的多级流水处理单元的执行流程示意图；

[0029] 图3是本申请另一实施例提供的一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法的流程示意图；

[0030] 图4是本申请实施例提供的一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的结构示意图；

[0031] 图5是本申请实施例提供的多级流水处理单元的结构示意图；

[0032] 图6是本申请另一实施例提供的一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的结构示意图；

[0033] 图7是本申请实施例提供的电子设备的系统构成示意框图。

具体实施方式

[0034] 为使本发明实施例的目的、技术方案和优点更加清楚明白，下面结合附图对本发明实施例做进一步详细说明。在此，本发明的示意性实施例及其说明用于解释本发明，但并不作为对本发明的限定。

[0035] 下面首先对本申请所涉及的技术术语进行简单说明：

[0036] 术语“分布式集群”：一种计算机系统架构，它将多台独立的计算机(节点)通过网络连接起来，共同执行任务或应用。分布式集群通过将计算任务分散到多个计算节点上，不仅提高了计算效率和系统的可靠性，还提供了优异的可扩展性和灵活性，成为处理现代大规模计算问题的重要技术基础。

[0037] 术语“宏指令处理器”：一种高级的处理器设计，它能够执行由多条微指令(microinstructions)或操作序列组成的复杂指令，这些复杂指令被称为宏指令(Macro-Instructions)。

[0038] 如图1所示为本申请实施例提供的一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法的流程示意图，该方法包括如下步骤：

[0039] 步骤S101：基于硬件逻辑检测消息缓存接口和信令缓存接口确定是否有新的消息或者信令输入。

[0040] 消息缓存(Message FIFO)接口是专门用于缓存从集群系统的其他节点或外部源接收到的消息，这些消息可以包括数据更新、处理请求或其他形式的信息交换。

[0041] 信令缓存(Command FIFO)接口是用于接收和缓存控制信令，这些信令用于管理和协调集群中的任务和资源，例如启动或停止任务的命令，资源分配指令等。

[0042] 消息缓存接口和信令缓存接口都是采用了先进先出的队列管理机制，确保数据按照其到达顺序被处理。在本实施例中，消息缓存接口中的消息包括了计算指令，而信令缓存接口中的信令则包括了通信指令。

[0043] 消息和信令都是以数据包的形式存在的，该数据包包括了数据负载和标头，其中数据负载是实际的用户数据或计算数据，是处理器执行计算任务的基础，而标头则包含了用于路由决策的元数据，如目标地址、源地址等，确保数据包能够被正确地发送到目标处理

器。非阻塞型宏指令多级流水处理器的每个核心都包括一个路由器、一个数据包管理器和本地存储,其中路由器负责决定数据包的转发路径,确保数据包沿最优路径到达目标位置;数据包管理器处理数据包的队列和优先级,管理数据包的接收和发送;本地存储提供了进行计算所需数据的临时或持久存储空间。这种设计允许每个处理器核心独立处理数据包的接收、计算和转发。

[0044] 在本实施例中,非阻塞型宏指令处理器可以定期检查消息缓存和信令缓存接口,以确定是否有新的输入等待处理,这种检测机制使处理器能够及时响应外部事件和内部指令。

[0045] 作为“非阻塞”的一部分,宏指令处理器可以根据当前的工作负载和系统状态动态调整其内部配置,例如调整资源分配,或改变指令执行的优先级,以优化处理性能。另外本申请是通过硬件逻辑来检测消息缓存接口和信令缓存接口,硬件逻辑允许处理器在处理核心计算任务的同时,独立地检测消息和信令接口,这意味着数据处理和信号检测可以同时发生,一方不会因为另一方的处理而等待或阻塞。而且本申请的消息和信令都是通过独立的数据通道进行传输的,本申请的非阻塞宏指令处理器能够在不中断主要计算任务的同时,高效地处理外部通信任务,这在多任务和高并发分布式集群环境中尤其重要,充分体现了非阻塞的设计理念。

[0046] 步骤S102:响应于有新的消息或者信令输入,检测消息或信令数据包中的任务标识符和目标端口号是否符合当前处理需求,若符合则直接从所述消息缓存接口或所述信令缓存接口读取消息或信令数据包到寄存器中并进入步骤S103,若不符合则丢弃。

[0047] 在本实施例中,每个消息或信令数据包都包含一个任务标识符(TASK_ID),该任务标识符唯一地标识了数据包关联的任务或作业,这允许处理器判断数据包是否属于当前正在处理或待处理的任务队列。数据包还指定了目标端口号(Channel),它表示数据包应该被路由到的特定程序的处理单元。在一个分布式集群系统中,不同的处理单元可以监听不同的端口号,处理特定类型的任务。

[0048] 假设本申请有一个宏指令执行器在分布式集群系统中,它需要处理8个处于激活(Active)状态的程序,每个程序分配有唯一的目标端口号,范围从0到7。宏指令执行器在启动时,会初始化其内部资源和配置,包括为每个目标端口号(0-7)分配必要的处理资源,每个激活状态的程序被分配一个唯一的目标端口号,这个目标端口号在程序的执行期间用于标识和路由其数据包和指令。

[0049] 宏指令执行器不断检测输入队列,以识别新的消息或信令数据包,每个数据包都包含目标端口号,表示它所关联的程序。根据数据包中的目标端口号,宏指令执行器将数据包路由到对应的处理队列,这确保了每个程序的数据包能够被正确地识别和处理。宏指令执行器可以同时处理所有活跃程序的数据包,由于每个程序有一个独立的处理队列和资源分配,这些程序可以并行执行,互不干扰。当程序完成其任务时,它的状态会被更新为非激活,宏指令执行器可以回收该程序所使用的资源,供其他任务使用。

[0050] 优选的,本申请的执行方法还包括了类似于中断处理程序的功能,即检测到新输入的消息为触发消息(Vector message)时,则根据消息内容和预先配置的偏移量,定位到对应的子程序区进行处理,且每个vector对应的子程序区不需要回调(Call)或者是返回(Return)指令,执行完毕后,可以直接转移到下一个任务或返回到被中断前的状态。这种类

似于中断处理程序的功能,提高了分布式集群系统处理异步事件的能力和效率。

[0051] 进一步优选的,本申请的执行方法还包括了仲裁过程,即当所述消息缓存接口和信令缓存接口为空的时候,在可用的端口号所对应的宏指令程序之间进行仲裁,确定一个端口号后将确定的端口号及启动命令发送给多级流水处理单元。

[0052] 当分布式集群系统中的消息缓存接口和信令缓存接口都为空时,表示当前没有待处理的外部消息或信令。在这种情况下,系统需要有效利用其计算资源,确保系统持续高效运行。通过在可用的端口号所对应的宏指令程序之间进行仲裁,系统可以选择接下来要执行的任务,这一过程的目的是保持处理单元的高效利用,并优化整体的计算资源分配。

[0053] 由于每个宏指令程序都与一个特定的端口号关联,这个端口号标识了程序的通信和数据流路径,仲裁过程即决定了哪一个端口号所对应的程序将被执行。在确定要执行的端口号后,系统生成一个启动命令,该命令包含了执行选定宏指令程序所需的所有信息,如起始地址、执行参数等。启动命令和确定的端口号被发送到多级流水处理单元,以此来启动宏指令程序的执行流程。系统通过这种方式动态调整其任务执行队列,确保即使在外输入缺乏时也能保持计算资源的有效利用。因此,该过程有助于优化系统的整体效率和响应性,通过内部任务的合理安排和执行,最大化资源的利用率和系统的吞吐量。

[0054] 步骤S103:将所述目标端口号和启动命令发送给多级流水处理单元进行取指、译码、取数、执行和写回处理。

[0055] 在本实施例中,将目标端口号和启动命令作为输入,通过多级流水处理单元的一系列处理,可以高效地实现指令的执行,特别是在高性能计算或者需要快速响应的应用场景中,流水线技术通过并行处理多个指令的不同阶段,显著提高了处理器的吞吐率和执行效率。

[0056] 优选的,本实施例的级流水处理单元包括取指模块、译码模块、取数模块、执行模块和写回模块,如图2所示,本步骤可以进一步包括如下子步骤:

[0057] 步骤S1031:将所述目标端口号和启动命令发送给取指模块,由所述取指模块根据所述目标端口号查询对应的指令地址寄存器以获得指令,并将所述指令发送给译码模块。

[0058] 以上述需要处理8个处于激活状态的程序为例,取指模块中可以有8个指令地址寄存器(Instruction address Reg)和8个堆栈指针寄存器(Stack Pointer Reg),若步骤S102中收到的消息为Reset消息的时候,消息所对应的指令地址寄存器与堆栈指针寄存器会被设置到初始值,这使得系统能够确保从一个已知、干净的状态开始执行。

[0059] 指令地址寄存器用于存储下一条要执行指令的内存地址,堆栈指针寄存器用于管理调用栈(call stack)或执行栈(execution stack)的状态。在收到目标端口号和启动命令之后,取指模块根据目标端口号查询对应的指令地址寄存器,从这个指令地址寄存器存储的地址取指并且交由后续模块执行。执行完一个指令Class之后,取指模块将对应的指令地址寄存器刷新到最新数值,然后将控制权交还给分布式集群系统的非阻塞型宏指令处理器。

[0060] 优选的,本实施例中的取指模块还支持异常处理和中断处理,即当所述取指模块检测到指令访存冲突的时候停止本级功能的执行,待冲突解决后再恢复本级功能的执行。

[0061] 指令访存冲突指的是指令在执行过程中遇到的内存访问冲突问题,这种冲突主要发生在处理器尝试同时访问内存中的多个位置,而这些访问操作之间存在数据依赖性或者资

源争用,导致无法同时满足所有内存访问请求。比如本实施中取数模块有读Ram操作,而写回模块有写Ram操作的时候,取指模块会停止本级功能的执行,待冲突解决后再恢复本级功能的执行。通过引入中断功能,非阻塞型宏指令多级流水处理器能够有效地管理和减轻指令访存冲突带来的影响,提高执行效率和处理器性能。

[0062] 优选的,当所述取指模块检测到指令访存冲突的时候,本申请的方法还包括:将访存冲突的指令进行暂时压栈处理,直到冲突解决后从栈中恢复所述访存冲突的指令并继续执行,这种方法可以有效避免由于访存冲突引起的长时间阻塞。

[0063] 优选的,本申请的方法还包括:根据所述消息或信令的当前执行情况和资源占用状态动态分配所述非阻塞型宏指令多级流水处理器的资源,比如可以采用优先级排队、时间片轮转或其他公平调度算法来确保各个指令平等且有效地访问所需的资源。此外,还可以引入一个反馈系统,根据指令执行的实时反馈调整资源分配策略,以提高资源利用率和执行效率。

[0064] 步骤S1032:所述译码模块接收到指令后进行初步译码得到指令码,并将所述指令码输出给取数模块,若所述译码模块遇到流水线停顿的状态,则进入延迟处理状态,并在下一个周期再输出译码结果。

[0065] 本申请译码模块的设计允许它在处理过程中动态适应当前的执行环境,以两种状态运行:Normal(正常状态)和Delay(延迟处理状态),这种设计增强了处理器对不同执行挑战的应对能力,特别是在面对流水线停顿时。

[0066] Normal状态下,译码模块接收来自取指模块的指令,这些指令已经从内存中取出,并准备进行译码,译码过程包括解析指令的指令码,确定指令的类型(例如,是算术操作、逻辑操作、还是数据传输操作等),并识别出操作数。

[0067] Delay状态被设计来应对流水线停顿的情况,流水线停顿可能由多种原因引起,如数据依赖、资源争夺、或者是由于前面的指令尚未完成执行而不能继续处理新的指令。当译码模块遇到这样的情况时,它会暂停当前的解码操作,并进入Delay状态。在这个状态下,指令的解码和传递被推迟到能够安全继续执行为止。这种设计允许流水线在维持数据完整性和执行正确性的前提下,灵活地处理执行过程中的各种延迟。

[0068] 因此,本申请的译码模块根据流水线的当前状态动态切换其操作状态,这种状态切换机制提高了处理器对执行障碍的适应性,帮助减少因停顿导致的性能损失。

[0069] 步骤S1033:所述取数模块根据指令码解析出执行模块和写回模块两级逻辑需要的操作码并且向后传输。

[0070] 取数模块将解析出的操作码和必要的操作信息传输给执行模块和写回模块,以指导它们完成具体的逻辑和写回操作。在本级逻辑中,取数模块同样会根据指令码执行读RAM或者是读reg操作。

[0071] 优选的,取数模块还用于接收所述写回模块产生的流水线延迟(pipeline_delay)、流水线丢弃(pipeline_discard)以及原子操作结束信号;

[0072] 当收到原子操作结束信号时,取数模块丢弃当前操作,并交还控制权给所述非阻塞型宏指令多级流水处理器,这里原子操作是指在执行过程中不可被其他线程中断的操作,在多线程环境下,原子操作可以在不需要锁定资源的情况下确保数据的一致性和完整性;

[0073] 当收到流水线丢弃信号时,取数模块丢弃当前操作,并继续执行后续指令;

[0074] 进一步优选的,上述取指模块和译码模块也可以接收所述写回模块产生的原子操作结束信号,并基于所述原子操作结束信号丢弃当前操作,然后交还控制权给所述非阻塞型宏指令多级流水处理器。

[0075] 这一复杂的信号处理和状态管理机制体现了非阻塞型宏指令多级流水处理器在指令执行、异常管理和任务调度方面的高度复杂性和灵活性。通过这种精细控制,非阻塞型宏指令多级流水处理器能够有效地应对执行过程中遇到的各种情况,保持高效率的运行,同时确保执行的正确性和系统的稳定性。

[0076] 进一步优选的,当写回模块产生原子操作结束信号后,非阻塞型宏指令多级流水处理器可以将当前状态进行暂存,接着接收并解析新的消息或信令输入,基于解析的结果由多级流水处理单元进行新一轮流水操作。通过这种方式,该非阻塞宏指令处理器能够有效地处理并行任务,提高执行效率和响应速度。它可以在完成一项任务的关键步骤后,立即转而处理新的输入,从而实现高效的多任务处理能力,这就是其非阻塞特征的具体实现方式。

[0077] 步骤S1034:所述执行模块基于所述操作码完成指令的执行。

[0078] 执行模块在多级流水线处理器架构中负责执行阶段,在这个阶段,处理器对前面阶段解析和准备好的指令执行具体的逻辑或算术操作。执行模块的工作不仅限于执行简单的算术运算,还涉及复杂的数据处理和逻辑判断:比如执行模块需要处理来自RAM的数据(Ram Data)以及来自寄存器的数据(Register Data),这包括从指定的内存地址读取数据和从寄存器文件中获取数据;还需要根据指令的要求,执行加法或减法运算,这些操作可以涉及处理整数或浮点数数据,且需要正确处理运算中产生的进位(carry)或借位(borrow);在需要将更新的数据写回RAM时,执行模块执行一个特定的操作序列:首先读取目标内存地址中的原始64位数据,以确保只修改目标数据段,而不影响其他数据;将需要写入的新数据(来自寄存器或计算结果)整合到读取的原始数据中,替换掉原有的相应段落或值;将经过更换的完整的64位数据写回到RAM的相同地址。另外,加减法运算产生的进位信号(carry flag)还可以用作条件判断,控制条件跳转指令的执行,例如,在某些条件分支指令中,仅当进位标志被设置(或未设置)时才执行跳转。

[0079] 本实施例中执行模块的设计体现了非阻塞型宏指令多级流水处理器在执行阶段的高度复杂性和灵活性,它能够处理各种数据操作,支持复杂的程序逻辑,并对指令执行过程中的条件变化作出响应。

[0080] 步骤S1035:所述写回模块基于所述执行模块的执行结果和所述操作码完成写入操作、栈操作以及控制流操作。

[0081] 写回模块是多级流水线处理器架构中的最后一个阶段,此阶段的主要职责是完成指令执行的最终步骤,将执行结果写回到目标位置,并处理指令中的控制流操作。

[0082] 其中,写入操作包括写RAM和写Reg:写RAM是将执行结果或处理后的数据写入到内存中的指定位置,这可以是算术运算的结果、数据处理操作的输出,或其他指令产生的数据;写Reg是更新寄存器文件中的特定寄存器,执行结果直接写入到一个或多个目标寄存器中,供后续指令使用。

[0083] 栈操作包括PUSH、POP:PUSH是将数据(如函数调用的参数、局部变量或返回地址)

压入调用栈;POP是从调用栈顶部移除数据,并可能将其加载到寄存器或内存中。

[0084] 控制流操作包括CALL、RETURN和跳转操作:CALL执行函数调用操作,包括将返回地址(即CALL指令之后的指令地址)保存到调用栈,并跳转到函数的起始地址执行。RETURN是从函数返回到调用点,这通常涉及从调用栈中弹出返回地址,并将程序计数器(PC)设置到该地址,以恢复调用前的执行流程。跳转操作是根据指令或执行条件改变程序的执行流,跳转可能是无条件的,也可能是基于特定条件(如比较操作的结果),在需要执行跳转时,写回模块生成一个跳转操作信号,并将此信号发送回取指模块,指示需要改变当前的执行路径,这要求取指模块根据接收到的信号更新指令地址寄存器,以指向新的指令地址。

[0085] 写回模块的设计和功​​能反映了处理器在完成指令执行、管理控制流和保持程序状态一致性方面的复杂性。通过精确地执行这些操作,写回模块确保了程序逻辑的正确实现,并为流水线处理器的高效运行提供了支持。

[0086] 优选的,上述各模块是在同一时钟周期协调工作,完成数据流和控制和传输,因各个模块在同一时钟周期内操作并行,即在同一时钟周期内模块执行的数据是没有数据相关性的,并且各个模块的执行与否可以由其控制指令来控制实现。这样系统能够在每个时钟周期处理更多的指令和数据,显著提高了处理速度和吞吐率。这种设计充分利用了硬件资源,提高了系统的整体效率。由于在同一时钟周期内,模块执行的数据没有数据相关性,这种设计有效地避免了数据冲突和依赖,减少了需要额外处理数据依赖性所导致的延迟和复杂度。每个模块可以独立处理其数据,无需等待其他模块的结果。最后,将系统设计为模块化且各模块能够在同一时钟周期内并行工作,使得系统更容易扩展和维护,新增功能或提升性能时,可以通过增加新的模块或优化现有模块来实现,而无需重构整个系统。

[0087] 步骤S104:利用独立的消息或信令逻辑接口封装并输出处理后的消息或者信令。

[0088] 由上述技术方案可知,本发明提供的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,将计算指令和通信指令进行了解耦,通过数据通道及信令通道分别传输计算指令和通信指令,并利用消息缓存接口和信令缓存接口分别接收计算指令和通信指令,基于控制逻辑单元和多级流水处理单元完成指令的并行处理,应用于分布式集群系统可以实现网络系统的通信效率提升。

[0089] 如图3为本申请实施例提供的另一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法的流程示意图,该方法包括如下步骤:

[0090] 步骤S301:接收任务源代码,并从源代码中确定能向量化的数据循环体。

[0091] 非阻塞型宏指令多级流水处理器首先接收到用于深度学习或其他计算密集型任务的执行源代码,通过特定的分析工具或编译器技术,可以从执行源代码中识别出可以进行向量化处理的数据循环体,这一步骤是在软件层面完成的,目的是优化代码的执行效率。

[0092] 步骤S302:对数据循环体的指令作向量化处理生成通信指令和计算指令。

[0093] 一旦确定了哪些循环体可以向量化,非阻塞型宏指令多级流水处理器会对这些循环体的指令进行向量化处理,生成相应的通信指令和存储指令。这些指令被设计为可以在非阻塞型宏指令多级流水处理器上高效执行。

[0094] 步骤S303:将通信指令和计算指令分别封装成单个数据包,每个数据包包括了数据负载和标头。

[0095] 通信指令和计算指令被封装成单个数据包,每个数据包包含数据负载和标头。这

些数据包在核心网格内进行识别和路由。每个非阻塞型宏指令多级流水处理器对数据包进行计算,核心包括路由器、数据包管理器和存储,支持片内和片外的同步协同。

[0096] 步骤S304:基于硬件逻辑检测消息缓存接口和信令缓存接口确定是否有新的消息数据包或者信令数据包输入。

[0097] 步骤S305:响应于有新的消息或者信令输入,检测消息或信令数据包中的任务标识符和目标端口号是否符合当前处理需求,若符合则直接从所述消息缓存接口或所述信令缓存接口读取消息或信令数据包到寄存器中并进入步骤S306,若不符合则丢弃。

[0098] 步骤S306:将所述目标端口号和启动命令发送给多级流水处理单元进行取指、译码、取数、执行和写回处理。

[0099] 步骤S307:利用独立的消息或信令逻辑接口封装并输出处理后的消息或者信令。

[0100] 由上述技术方案可知,本发明提供的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行方法,将计算指令和通信指令进行了解耦,通过数据通道及信令通道分别传输计算指令和通信指令,并利用消息缓存接口和信令缓存接口分别接收计算指令和通信指令,基于控制逻辑单元和多级流水处理单元完成指令的并行处理,从而将计算指令和通信指令操作和执行进行了解耦,应用于分布式集群系统可以实现网络系统的通信效率提升。

[0101] 如图4所示为本申请实施例提供的一种用于分布式集群系统的非阻塞型宏指令多级流水处理器的结构示意图,其包括:控制逻辑单元410、多级流水处理单元420和输出单元430,它们之间依次相连,其中:

[0102] 控制逻辑单元410,用于基于硬件逻辑检测消息缓存接口和信令缓存接口确定是否有新的消息或者信令输入;响应于有新的消息或者信令输入,检测消息或信令数据包中的任务标识符和目标端口号是否符合当前处理需求,若符合则直接从所述消息缓存接口或所述信令缓存接口读取消息或信令数据包到寄存器中并将所述目标端口号和启动命令发送给多级流水处理单元420,若不符合则丢弃。

[0103] 多级流水处理单元420,用于基于所述目标端口号和启动命令进行取指、译码、取数、执行和写回处理。

[0104] 输出单元430,用于利用独立的消息或信令逻辑接口封装并输出处理后的消息或者信令。

[0105] 优选的,上述控制逻辑单元410还用于:若检测到新输入的消息为触发消息,则根据消息内容和预先配置的偏移量,定位到对应的子程序区进行处理。

[0106] 优选的,上述控制逻辑单元410还用于:当所述消息缓存接口和信令缓存接口为空的时候,在可用的端口号所对应的宏指令程序之间进行仲裁,确定一个端口号后将确定的端口号及启动命令发送给所述多级流水处理单元。

[0107] 优选的,如图5所示,上述多级流水处理单元420包括取指模块421、译码模块422、取数模块423、执行模块424和写回模块425,其中取指模块421用于根据所述目标端口号查询对应的指令地址寄存器以获得指令,并将所述指令发送给译码模块422;所述译码模块422用于接收到指令后进行初步译码得到指令码,并将所述指令码输出给取数模块423,若所述译码模块422遇到流水线停顿的状态,则进入延迟处理状态,并在下一个周期再输出译码结果;所述取数模块423用于根据指令码解析出执行模块和写回模块两级逻辑需要的操

作码并且向后传输;所述执行模块424用于基于所述操作码完成指令的执行;所述写回模块425用于基于所述执行模块的执行结果和所述操作码完成写入操作、栈操作以及控制流操作。

[0108] 优选的,上述取指模块421还用于检测到指令访存冲突的时候停止本级功能的执行,待冲突解决后再恢复本级功能的执行。

[0109] 优选的,当所述取指模块检测到指令访存冲突的时候,所述装置还包括压栈处理单元:用于将访存冲突的指令进行暂时压栈处理,直到冲突解决后从栈中恢复所述访存冲突的指令并继续执行。

[0110] 优选的,所述装置还包括资源调度单元:用于根据所述消息或信令的当前执行情况 and 资源占用状态动态分配所述非阻塞型宏指令多级流水处理器的资源。

[0111] 优选的,上述取数模块423还用于接收所述写回模块425产生的流水线延迟、流水线丢弃以及原子操作结束信号;当收到原子操作结束信号时,所述取数模块423丢弃当前操作,并交还控制权给所述非阻塞型宏指令多级流水处理器;当收到流水线丢弃信号时,所述取数模块423丢弃当前操作,并继续执行后续指令;所述取指模块421和所述译码模块422还用于接收所述写回模块425产生的原子操作结束信号,并基于所述原子操作结束信号丢弃当前操作,然后交还控制权给所述非阻塞型宏指令多级流水处理器。

[0112] 优选的,当所述写回模块产生原子操作结束信号后,所述非阻塞型宏指令多级流水处理器将当前状态进行暂存,接着接收并解析新的消息或信令输入,基于解析的结果由多级流水处理单元进行新一轮流水操作。

[0113] 进一步的,上述用于分布式集群系统的非阻塞型宏指令多级流水处理器的结构也可以参见图6,其中多级流水处理单元420各个模块在执行各自功能的时候还涉及相关数据段、信令段、虚转换段的输入输出,即本申请的非阻塞宏指令处理器具有指令缓存和数据缓存和虚实转换缓存和存储系统,可以为多级流水处理单元420提供处理所需的相关数据段、信令段、虚转换段。

[0114] 由上述技术方案可知,本发明提供的用于分布式集群系统的非阻塞型宏指令多级流水处理器的执行装置,将计算指令和通信指令进行了解耦,通过数据通道及信令通道分别传输计算指令和通信指令,并利用消息缓存接口和信令缓存接口分别接收计算指令和通信指令,基于控制逻辑单元和多级流水处理单元完成指令的并行处理,应用于分布式集群系统可以实现网络系统的通信效率提升。

[0115] 本发明实施例还提供一种电子设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,处理器执行所述程序时实现上述方法。

[0116] 本发明实施例还提供一种计算机可读存储介质,计算机可读存储介质存储有执行上述方法的计算机程序。

[0117] 如图7所示,该电子设备600还可以包括:通信模块110、输入单元120、音频处理器130、显示器160、电源170。值得注意的是,电子设备600也并不是必须要包括图7中所示的所有部件;此外,电子设备600还可以包括图7中没有示出的部件,可以参考现有技术。

[0118] 如图7所示,中央处理器100有时也称为控制器或操作控件,可以包括微处理器或其他处理器装置和/或逻辑装置,该中央处理器100接收输入并控制电子设备600的各个部件的操作。

[0119] 其中,存储器140,例如可以是缓存器、闪存、硬驱、可移动介质、易失性存储器、非易失性存储器或其它合适装置中的一种或更多种。可储存上述与失败有关的信息,此外还可存储执行有关信息的程序。并且中央处理器100可执行该存储器140存储的该程序,以实现信息存储或处理等。

[0120] 输入单元120向中央处理器100提供输入。该输入单元120例如为按键或触摸输入装置。电源170用于向电子设备600提供电力。显示器160用于进行图像和文字等显示对象的显示。该显示器例如可为LCD显示器,但并不限于此。

[0121] 该存储器140可以是固态存储器,例如,只读存储器(ROM)、随机存取存储器(RAM)、SIM卡等。还可以是这样的存储器,其即使在断电时也保存信息,可被选择性地擦除且设有更多数据,该存储器的示例有时被称为EPROM等。存储器140还可以是某种其它类型的装置。存储器140包括缓冲存储器141(有时被称为缓冲器)。存储器140可以包括应用/功能存储部142,该应用/功能存储部142用于存储应用程序和功能程序或用于通过中央处理器100执行电子设备600的操作的流程。

[0122] 存储器140还可以包括数据存储部143,该数据存储部143用于存储数据,例如联系人、数字数据、图片、声音和/或任何其他由电子设备使用的数据。存储器140的驱动程序存储部144可以包括电子设备的用于通信功能和/或用于执行电子设备的其他功能(如消息传送应用、通讯录应用等)的各种驱动程序。

[0123] 通信模块110即为经由天线111发送和接收信号的发送机/接收机110。通信模块(发送机/接收机)110耦合到中央处理器100,以提供输入信号和接收输出信号,这可以和常规移动通信终端的情况相同。

[0124] 基于不同的通信技术,在同一电子设备中,可以设置有多个通信模块110,如蜂窝网络模块、蓝牙模块和/或无线局域网模块等。通信模块(发送机/接收机)110还经由音频处理器130耦合到扬声器131和麦克风132,以经由扬声器131提供音频输出,并接收来自麦克风132的音频输入,从而实现通常的电信功能。音频处理器130可以包括任何合适的缓冲器、解码器、放大器等。另外,音频处理器130还耦合到中央处理器100,从而使得可以通过麦克风132能够在本机上录音,且使得可以通过扬声器131来播放本机上存储的声音。

[0125] 本领域内的技术人员应明白,本发明的实施例可提供为方法、系统、或计算机程序产品。因此,本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且,本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0126] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器,使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0127] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中,使得存储在该计算机可读存储器中的指令产生包括指

令装置的制造品,该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0128] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上,使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理,从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0129] 本发明中应用了具体实施例对本发明的原理及实施方式进行了阐述,以上实施例的说明只是用于帮助理解本发明的方法及其核心思想;同时,对于本领域的一般技术人员,依据本发明的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本发明的限制。

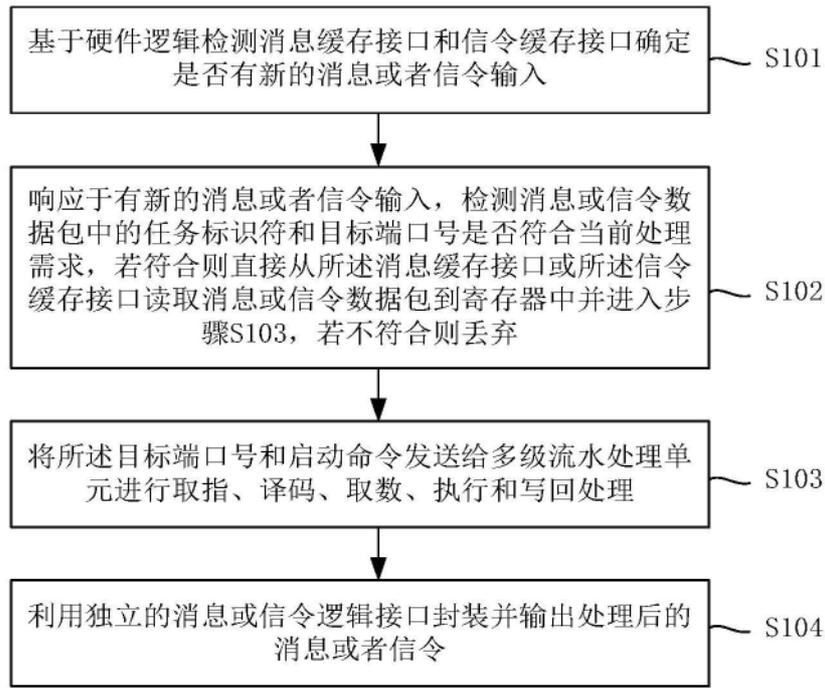


图1

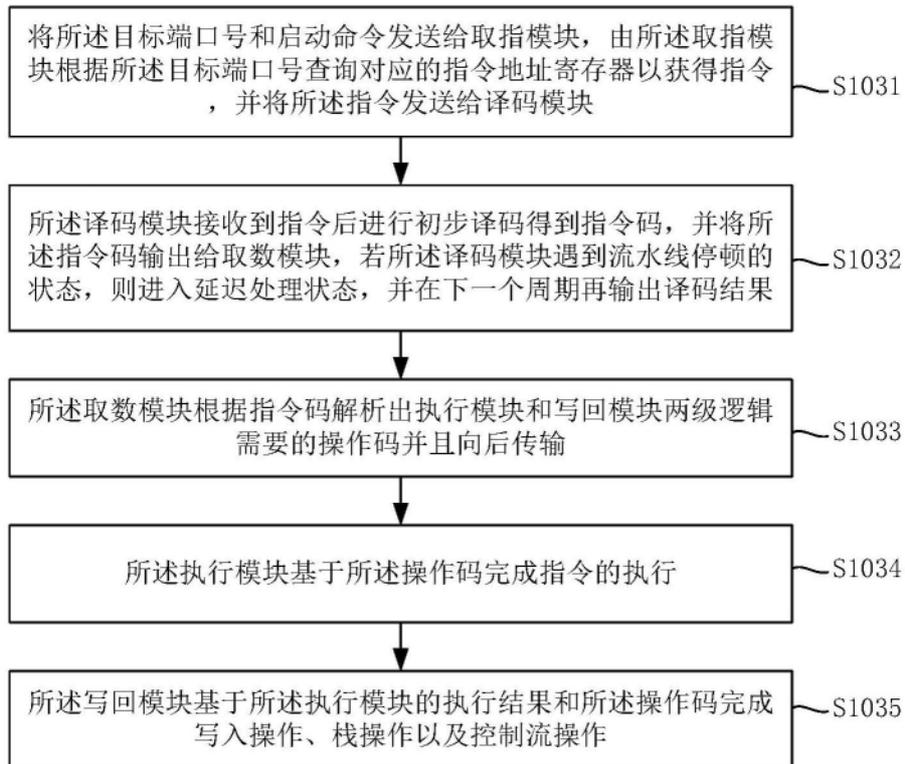


图2

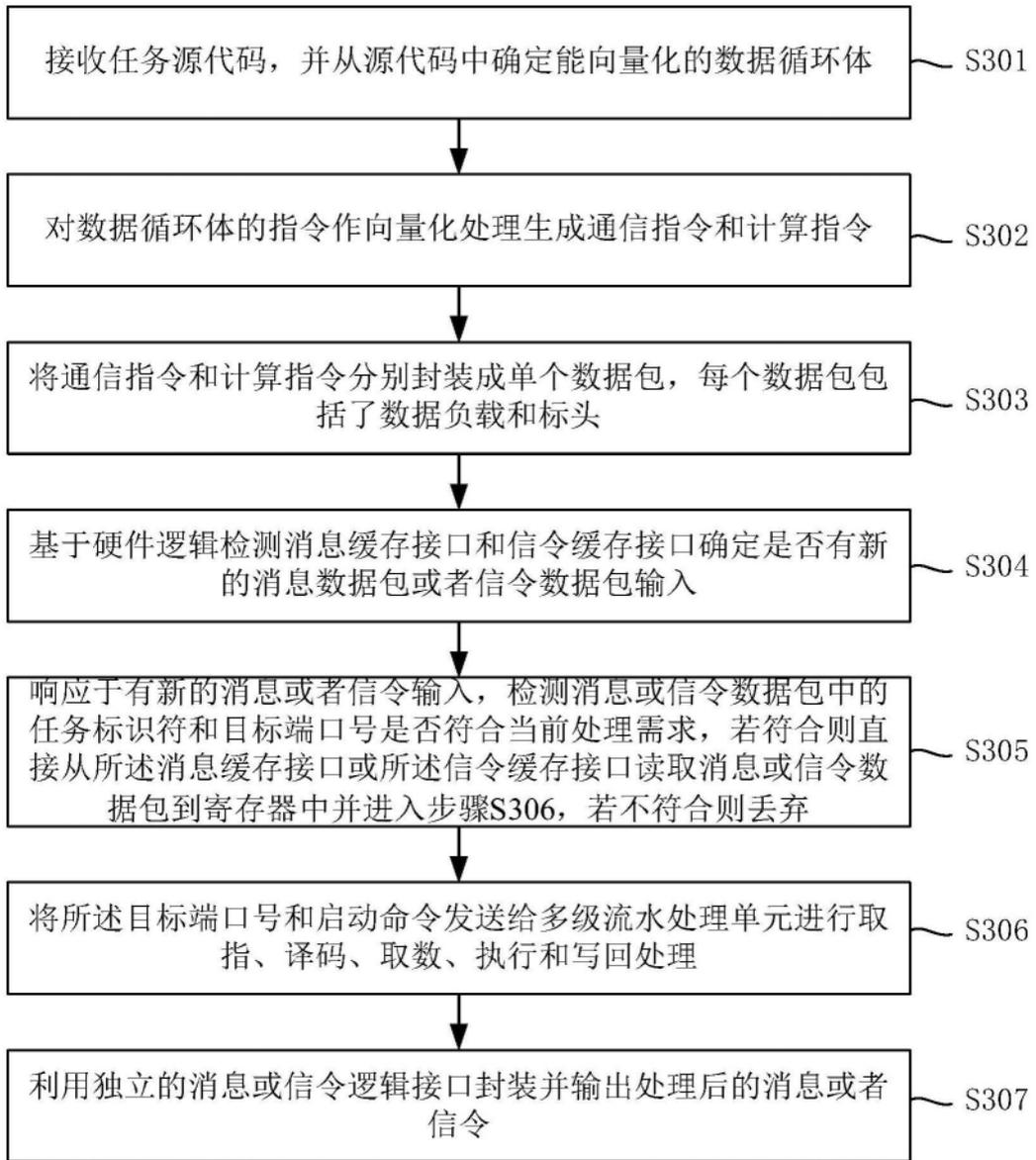


图3

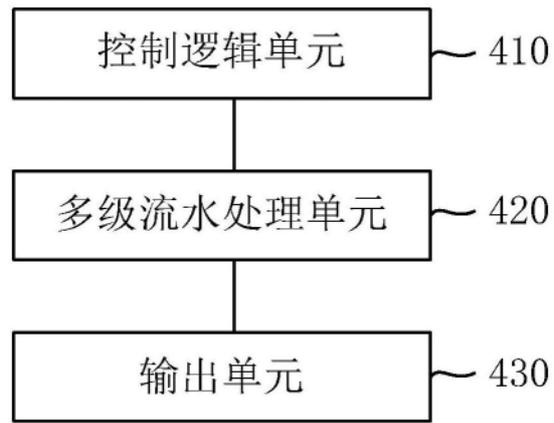


图4

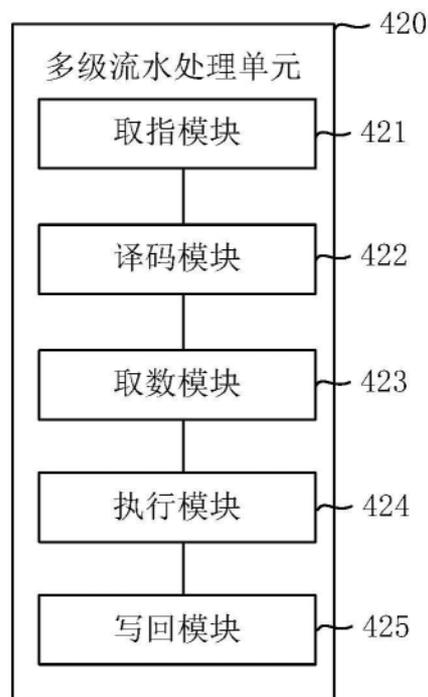


图5

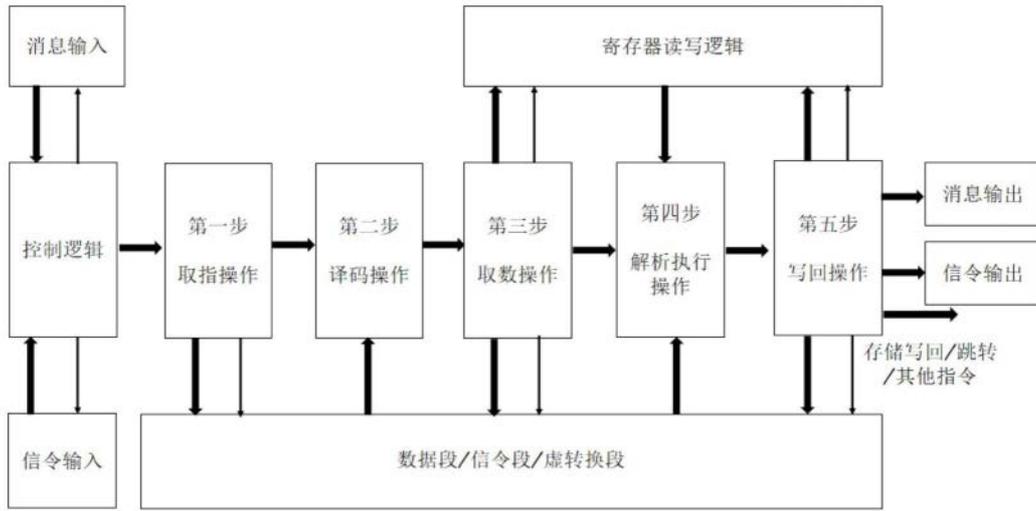


图6

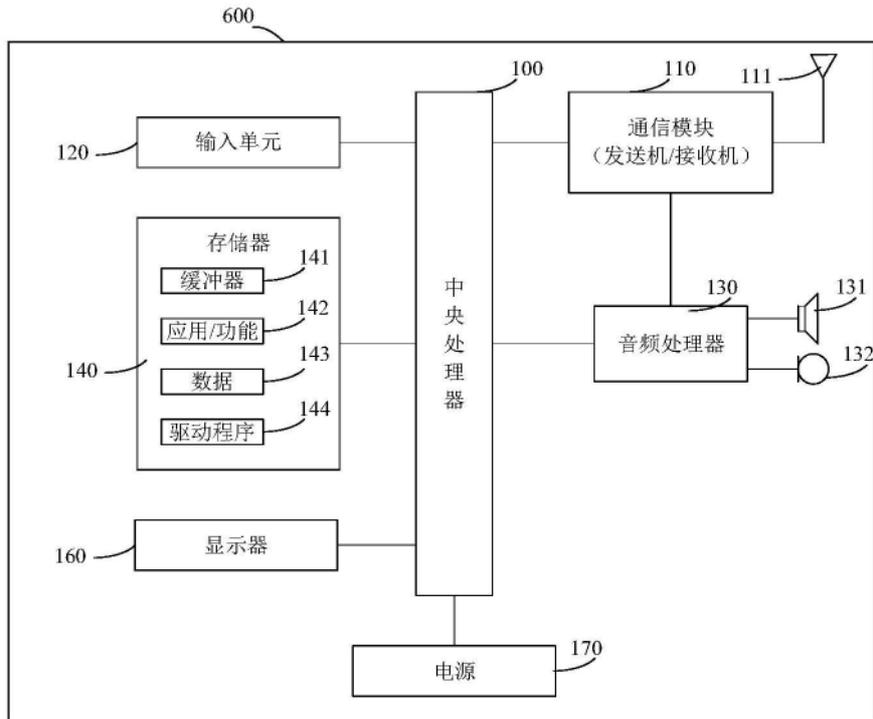


图7