



(12) 发明专利

(10) 授权公告号 CN 108762810 B

(45) 授权公告日 2021.01.08

(21) 申请号 201711447386.0

(22) 申请日 2017.12.27

(65) 同一申请的已公布的文献号
申请公布号 CN 108762810 A

(43) 申请公布日 2018.11.06

(73) 专利权人 北京时代民芯科技有限公司
地址 100076 北京市丰台区东高地四营门
北路2号

专利权人 北京微电子技术研究所

(72) 发明人 陶淑婷 赵沛 闫攀 毛雅欣
牛建泽

(74) 专利代理机构 中国航天科技专利中心
11009

代理人 庞静

(51) Int.Cl.

G06F 9/38 (2006.01)

H04L 12/743 (2013.01)

(56) 对比文件

CN 106612236 A, 2017.05.03

CN 101276294 A, 2008.10.01

CN 1858705 A, 2006.11.08

审查员 赵丽敬

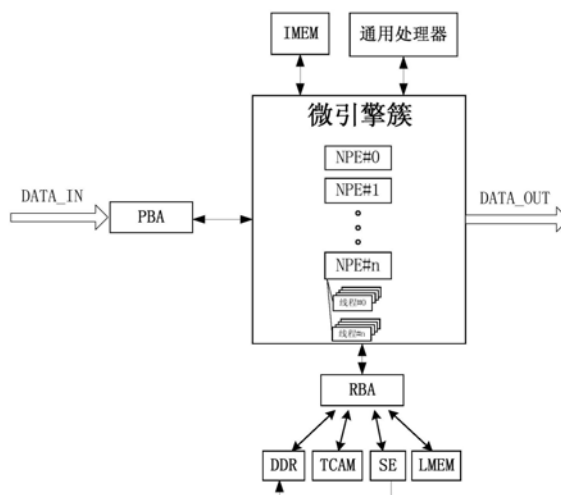
权利要求书1页 说明书4页 附图2页

(54) 发明名称

一种基于并行微引擎的网络报文头处理器

(57) 摘要

本发明公开了一种基于并行微引擎的网络报文头处理器,包括微引擎簇、轮询调度器模块、任务调度器模块、存储器模块,微引擎簇,由多个并行独立工作的微引擎组成,每个微引擎加载相应的微码指令,根据微码指令,调度多个线程通过任务调度器模块访问存储器模块中相应存储单元中的相关表项,完成报文头数据帧解析、分类和转发处理,并将各线程工作状态反馈给轮询调度器;轮询调度器模块,轮询各微引擎每个线程的线程工作状态,将收到的报文头递交给线程空闲数较多的微引擎;存储器模块,用于存储报文头数据帧解析、分类和转发处理所需要的相关表项和微引擎的微码指令。本发明可保证报文的高速处理和高度灵活性,满足网络大规模数据处理的需求。



1. 一种基于并行微引擎的网络报文头处理器,其特征在于包括微引擎簇、轮询调度器模块、任务调度器模块、存储器模块,其中:

微引擎簇,由多个并行独立工作的微引擎组成,每个微引擎加载相应的微码指令,根据微码指令,调度多个线程通过任务调度器模块访问存储器模块中相应存储单元中的相关表项,完成报文头数据帧解析、分类和转发处理,并将各线程工作状态反馈给轮询调度器;

轮询调度器模块,轮询各微引擎每个线程的线程工作状态,将收到的报文头递交给线程空闲数较多的微引擎;

存储器模块,用于存储报文头数据帧解析、分类和转发处理所需要的相关表项和微引擎的微码指令。

2. 根据权利要求1所述的一种基于并行微引擎的网络报文头处理器,其特征在于,所述任务调度器模块,调度各微引擎线程对存储器中的存储单元的访问,使得微引擎的线程之间以轮转非抢占方式对存储单元进行访问,以实现存储单元中的相关表项的查找和读写。

3. 根据权利要求1所述的一种基于并行微引擎的网络报文头处理器,其特征在于所述每个微引擎内部的线程之间采用流水线工作方式工作。

4. 根据权利要求1所述的一种基于并行微引擎的网络报文头处理器,其特征在于所述存储器模块中的存储单元包括DDR存储器、TCAM存储器、存片上存储器LMEM、指令存储器IMEM,其中:

DDR存储器,用于存储VLAN表、MPLS表;

TCAM存储器,用于存放MAC地址表、路由表;

片上存储器LMEM,用于存放流表;

指令存储器IMEM,用来存放微引擎的微码指令。

5. 根据权利要求4所述的一种基于并行微引擎的网络报文头处理器,其特征在于所述MAC地址表、路由表采用TCAM形式存储。

6. 根据权利要求4所述的一种基于并行微引擎的网络报文头处理器,其特征在于还包括搜索引擎,当微引擎访问DDR存储器时,微引擎通过任务调度器调用搜索引擎,并指定搜索引擎采用相应的搜索算法对DDR中的表项进行搜索,查找与微引擎所处理的报文头相匹配的表项,并将搜索结果反馈给微引擎。

7. 根据权利要求1所述的一种基于并行微引擎的网络报文头处理器,其特征在于微引擎簇中多个微引擎集成在一块芯片上。

8. 根据权利要求7所述的一种基于并行微引擎的网络报文头处理器,其特征在于所述芯片内部设有专门针对网络数据包处理的专用指令集,所述专用指令集包括乘法指令、循环冗余校验指令、按内容寻址指令、FFS指令,微引擎按照微码指令,调度线程执行这些指令,完成相应报文处理。

9. 根据权利要求1~8任一项所述的一种基于并行微引擎的网络报文头处理器,其特征在于还包括通用处理器,所述通用处理器用于使用标准编程语言编写微引擎的程序,经过编译形成微码指令,并将其下载到存储器模块中,上电后微引擎簇内的微引擎加载相应的微码指令,并按照微码指令调用线程工作,实现微引擎的重构。

一种基于并行微引擎的网络报文头处理器

技术领域

[0001] 本发明涉及一种基于并行微引擎的网络报文头处理器,属于有线通信技术领域。

背景技术

[0002] 目前市场上大多数网络处理器芯片的微引擎设计都采用流水线模式。在流水线模式中,数据包的处理分成若干个阶段,每个阶段都有固定的任务,数据包的处理过程就是从一个处理阶段跳到下一个处理阶段,直到处理完毕。每个流水阶段都包含一个或多个处理引擎,每个面向任务的处理引擎都是针对特定的网络任务以及它本身的指令集进行优化的。流水线体系结构基于一种分布式的编程模式,在这种模式中,应用程序分成多个流水阶段。每个流水线处理引擎都经过优化,以处理特定的任务和指令。这种模式的不足在于它需要把整个处理任务均匀的分配到流水线的各个阶段,否则流水线将会被打乱,且流水线模式下,每个微引擎之间相互影响,各微引擎闲忙不均,利用率较低,报文处理速度慢,无法满足未来网络大规模数据处理的需求。

发明内容

[0003] 本发明的技术解决问题是:克服现有技术的不足,提供一种基于并行微引擎的网络报文头处理器,并行微引擎之间并行独立工作,相互之间没有交互,内部对数据进行端对端并行处理,一个微引擎完成数据帧的整个流水线处理。这种并行模式可保证报文的高速处理,可满足未来网络大规模数据处理的需求。

[0004] 本发明的技术解决方案是:一种基于并行微引擎的网络报文头处理器,包括微引擎簇、轮询调度器模块、任务调度器模块、存储器模块,其中:

[0005] 微引擎簇,由多个并行独立工作的微引擎组成,每个微引擎加载相应的微码指令,根据微码指令,调度多个线程通过任务调度器模块访问存储器模块中相应存储单元中的相关表项,完成报文头数据帧解析、分类和转发处理,并将各线程工作状态反馈给轮询调度器;

[0006] 轮询调度器模块,轮询各微引擎每个线程的线程工作状态,将收到的报文头递交给线程空闲数较多的微引擎;

[0007] 存储器模块,用于存储报文头数据帧解析、分类和转发处理所需要的相关表项和微引擎的微码指令。

[0008] 所述任务调度器模块,调度各微引擎线程对存储器中的存储单元的访问,使得微引擎的线程之间以轮转非抢占方式对存储单元进行访问,以实现存储单元中的相关表项的查找和读写。

[0009] 所述每个微引擎内部的线程之间采用流水线工作方式工作。

[0010] 所述存储器模块中的存储单元包括DDR存储器、TCAM存储器、存片上存储器LMEM、指令存储器IMEM,其中:

[0011] DDR存储器,用于存储VLAN表、MPLS表;

[0012] TCAM存储器,用于存放MAC地址表、路由表;

[0013] 片上存储器LMEM,用于存放流表;

[0014] 指令存储器IMEM,用来存放微引擎的微码指令。

[0015] 所述MAC地址表、路由表采用TCAM形式存储。

[0016] 网络报文头处理器还包括搜索引擎,当微引擎访问DDR存储器时,微引擎通过任务调度器调用搜索引擎,并指定搜索引擎采用相应的搜索算法对DDR中的表项进行搜索,查找与微引擎所处理的报文头相匹配的表项,并将搜索结果反馈给微引擎。

[0017] 微引擎簇中多个微引擎集成在一块芯片上。

[0018] 所述芯片内部设有专门针对网络数据包处理的专用指令集,所述专用指令集包括乘法指令、循环冗余校验指令、按内容寻址指令、FFS指令,微引擎按照微码指令,调度线程执行这些指令,完成相应报文处理。

[0019] 网络报文头处理器还包括通用处理器,所述通用处理器用于使用标准编程语言编写微引擎的程序,经过编译形成微码指令,并将其下载到存储器模块中,上电后微引擎簇内的微引擎加载相应的微码指令,并按照微码指令调用线程工作,实现微引擎的重构。

[0020] 本发明与现有技术相比的有益效果是:

[0021] (1)、本发明的微引擎簇包含多个微引擎,每个微引擎都是独立并行执行的,微引擎之间没有任何关联,一个微引擎处理器完成数据帧的整个流水线处理,这种并行模式可保证报文的高速处理;

[0022] (2)、本发明的微引擎是可编程的,通过在通用处理器中使用高级语言进行应用程序开发,经过编译后,形成微码指令,下载到存储器模块上,上电后微引擎簇内的微微引擎加载相应的微码指令,按照微码指令调用线程对数据包进行相应的解析、处理和转发;

[0023] (3)、本发明的多个微引擎共享存储器,通过任务调度器可避免多个线程同时访问同一个存储器;

[0024] (4)、本发明包含一个轮询调度器模块,负责把数据包分配到某个微处理引擎,微引擎将各自线程的工作状态通过状态字反馈给轮询调度器,轮询调度器将报文分配给空闲数目较多的线程,以使各微引擎负载均衡。

附图说明

[0025] 图1为本发明一种基于并行微引擎的网络报文头处理器组成框图;

[0026] 图2为本发明实施例微引擎簇中的一个微引擎内部流水线结构设计;

[0027] 图3为本发明实施例中任务调度模块的实现。

具体实施方式

[0028] 下面就结合附图对本发明做进一步介绍。

[0029] 本发明公开了一种基于并行微引擎的网络报文头处理器,如图1所示,一种基于并行微引擎的网络报文头处理器包括微引擎簇、轮询调度器模块、任务调度器模块、搜索引擎、存储器模块、通用处理器,其中:

[0030] (1)、微引擎簇

[0031] 微引擎簇由多个并行独立工作的微引擎NPE(Network Processing Engine)组成,

每个微引擎包含多个线程,线程数目可根据实际需要通过对微码指令进行配置,一般可以配置为4线程或者8线程工作,微引擎簇工作过程中尽量保证每个微引擎的工作的线程数目接近。各微引擎并行独立工作,相互之间没有交互,每个微引擎完成一个报文的完整的处理流程,包括数据帧解析、分类和转发处理。

[0032] 所述微引擎内部的线程之间采用流水线工作模式工作,每个微引擎加载相应的微码指令,根据微码指令,调度多个线程通过任务调度器模块访问存储器模块中相应存储单元中的相关表项,完成报文头数据帧解析、分类和转发处理,并将各线程工作状态反馈给轮询调度器。多个微引擎集成在一块芯片上,芯片上包括有专门针对数据包处理的专用指令集,所述指令集内包括乘法指令、循环冗余校验指令、按内容寻址指令、FFS指令,微引擎通过执行这些专用指令,可以快速处理数据,提高性能。微引擎的线程将工作状态通过状态字反馈给轮询调度器,可以用1个bit表示线程的忙闲状态,1表示“忙”,0表示“闲”。通过多个微引擎的并行处理,微引擎簇可以实现高速、大容量智能数据帧处理。

[0033] 如图2所示为微引擎内部流水线结构设计。流水线由多个数据包指令功能块构成,它拥有专用指令集、代码数据存储资源、寄存器资源和处理控制权。每个线程对应一个数据包指令功能块,所有数据包指令功能块线性排列,每个数据包指令功能块执行一条指令后将数据包传递给下一个功能块,数据和状态由数据包上下文传递。

[0034] (2) 轮询调度器模块PBA(Packet Bus Allocation)

[0035] 轮询各微引擎线程的线程工作状态,根据其忙闲状态,并决定递交报文至哪个微引擎,通过统计每个微引擎的线程工作情况,将收到的报文递交给线程空闲数较多处理任务较少的微引擎,即将报文分派给负载最轻的微引擎。

[0036] (3) 任务调度器模块RBA(Resource Bus Allocation)

[0037] 任务调度器模块,完成总线调度功能,用于调度各线程对相关处理单元的访问,防止多个线程同时访问同一处理单元,造成访问冲突。微引擎的线程之间以轮转非抢占方式对相关存储单元(DDR(Double Data Rate)、TCAM(Ternary Content Addressable Memory)、SE(Search Engine)和IMEM(Instruction Memory))进行访问,以实现相关表项(MAC地址表、IP(Internet Protocol)路由表、VLAN(Virtual Local Area Network)表、MPLS表、流表等)的查找和读写。

[0038] 如图3所示为任务调度器处理设计。任务调度器的具体实现为:记录所有微引擎已处于准备访问存储器中的存储单元状态的线程号及其需要访问的存储单元,轮询该存储单元是否正处于被访问状态,当有线程完成对该存储单元的访问时,在记录的线程号中顺序搜寻一个准备访问该存储单元的线程,将访问权交给该线程。

[0039] 如图3所示,任务调度器从NPE1的线程1开始轮询处于Ready=1状态的线程,如微引擎NPE1的线程1和NPE2的线程2同时需要访问TCAM,此时如TCAM正在被其他线程访问,则NPE1的线程1和NPE2的线程2都处于等待状态,当TCAM被访问完毕后,任务调度器会先将TCAM的访问权交给NPE1的线程1,NPE2的线程2处于继续等待状态,直到NPE1的线程1释放TCAM的访问权。但TCAM在被访问时,如有其他线程需要访问DDR,此时DDR又没有其他线程访问,则任务调度器会将DDR的访问权交给该线程。

[0040] (4) 存储器模块

[0041] 存储器模块,包括多个存储单元,用于存储报文头数据帧解析、分类和转发处理所

需要的相关表项和微引擎的微码指令。

[0042] 存储器模块包括DDR存储器、TCAM、片上存储器LMEM、指令存储器IMEM。其中：

[0043] DDR存储器，用于存储VLAN表、MPLS表等业务相关且对处理速度要求相对较低的表项；微引擎通过任务调度器调用搜索引擎，并指定搜索引擎采用相应的搜索算法对DDR中的表项进行搜索，查找与微引擎所处理的报文头相匹配的表项，并将搜索结果反馈给微引擎。

[0044] TCAM存储器，用于存放MAC地址表、路由表等对处理速度要求较高的表项。所述MAC地址表、路由表采用TCAM形式存储，查找时，任务调度模块将报文头中信息，转换成TCAM表存储，与MAC地址表、路由表匹配，找到所需要的数据匹配项反馈给微引擎。

[0045] 片上存储器LMEM，用于存放流表，直接由微引擎的线程通过任务调度器进行访问。

[0046] 指令存储器IMEM，用来存放微引擎的微码指令。

[0047] (5) 搜索引擎

[0048] 用于DDR中表项的查找，可采用二分查找、树表查找、哈希查找等算法进行表项查找。

[0049] 搜索引擎SE为硬件协处理器，由硬件实现，微引擎内线程通过搜索引擎SE访问DDR中的数据表项，采用相应的搜索算法，如采用散列算法或二叉树搜索算法，查找与之相匹配的表项，并将执行结果返回给微引擎。

[0050] (6) 通用处理器

[0051] 通用处理器所述通用处理器用于使用标准编程语言编写微引擎的程序，经过编译形成微码指令，并将其下载到存储器模块中，上电后微引擎簇内的微引擎加载相应的微码指令，并按照微码指令调用线程工作，实现微引擎的重构。

[0052] 通用处理器还可以用来运行操作系统、芯片的初始化配置、对数据包做深层处理(4层协议以上处理)等。

[0053] 本发明根据不同的服务要求，可对包(帧)进行不同深度的处理。例如路由查找只需处理第3层(IP头部)，分类需要处理到第四层(TCP(Transmission Control Protocol)/UDP(User Datagram Protocol)协议)，而网络安全则需要处理到应用层(包携带的有效载荷)，对于4层以上的协议处理由通用处理器完成。

[0054] 本说明书中未进行详细描述部分属于本领域技术人员公知常识。

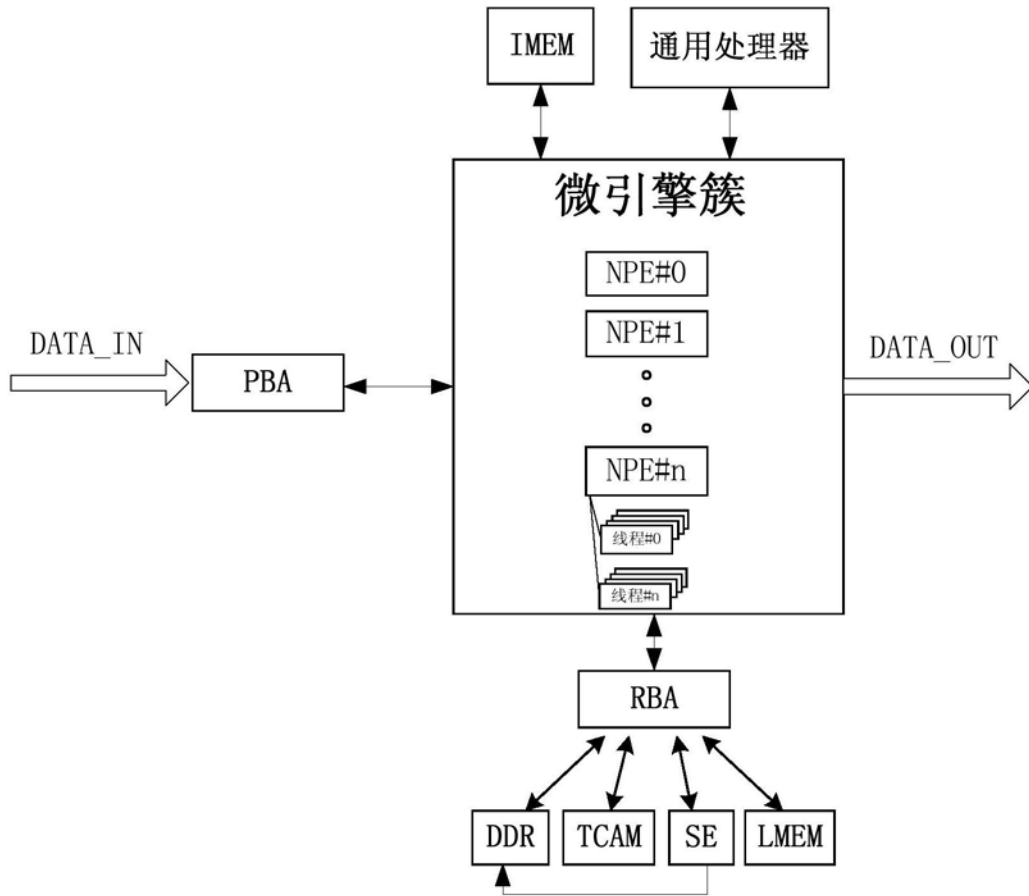


图1

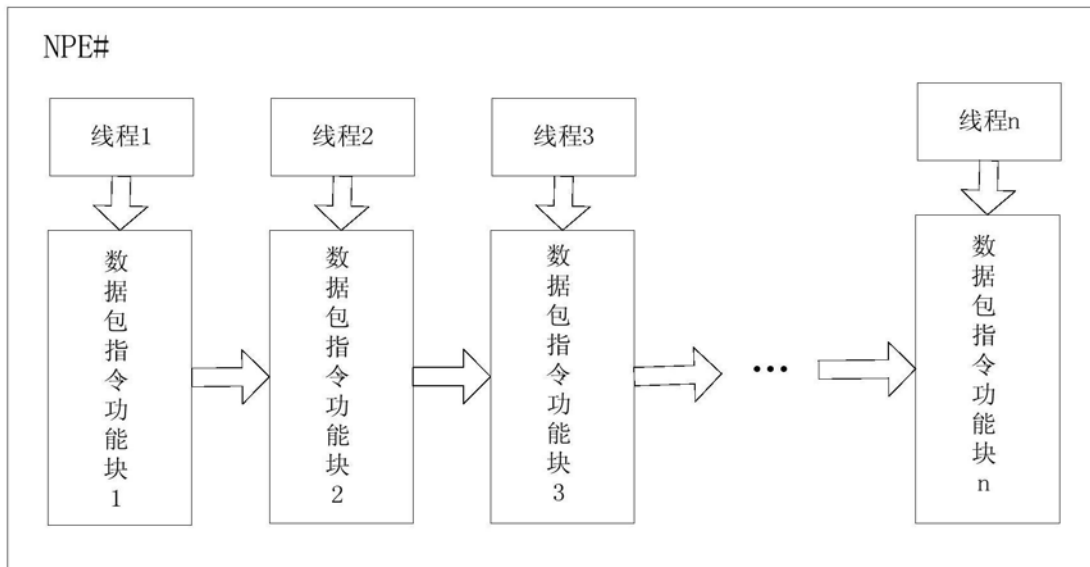


图2

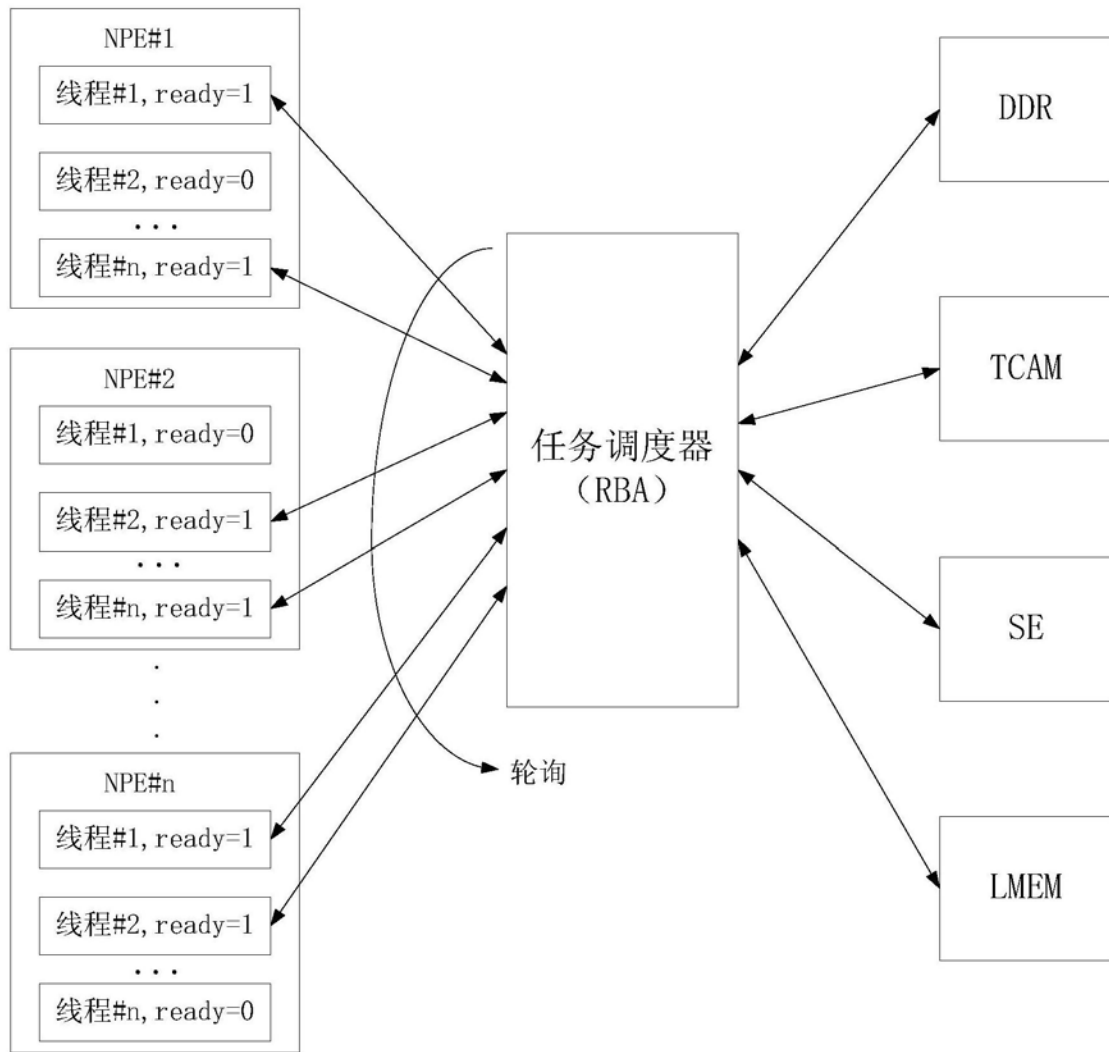


图3