



(12) 发明专利

(10) 授权公告号 CN 113687986 B

(45) 授权公告日 2024.09.13

(21) 申请号 202111013632.8

(22) 申请日 2021.08.31

(65) 同一申请的已公布的文献号

申请公布号 CN 113687986 A

(43) 申请公布日 2021.11.23

(73) 专利权人 上海阡视科技有限公司

地址 200232 上海市徐汇区丰谷路315弄24号1-3层

(72) 发明人 林志远 殷文达

(74) 专利代理机构 北京博思佳知识产权代理有限公司 11415

专利代理师 靳玫

(51) Int. Cl.

G06F 11/14 (2006.01)

(56) 对比文件

CN 105302768 A, 2016.02.03

CN 110569148 A, 2019.12.13

审查员 吴荻

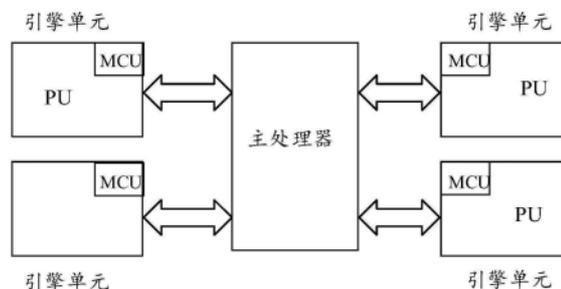
权利要求书2页 说明书9页 附图3页

(54) 发明名称

一种芯片和处理单元的恢复方法

(57) 摘要

本公开实施例提供一种芯片和处理单元的恢复方法,其中芯片包括:主处理器和多个引擎单元,每个引擎单元包括处理单元和用于控制处理单元的微控制单元;所述主处理器,用于接收所述引擎单元中的微控制单元发送的处理单元运行信息;响应于所述处理单元运行信息表示所述处理单元运行异常,向所述微控制单元发送恢复指令,所述恢复指令用于指示所述微控制单元恢复所述处理单元;各个所述微控制单元,用于响应于接收到所述主处理器发送的恢复指令,恢复所述处理单元。该芯片可以对单路发生异常情况的处理单元进行自动恢复,不再需要重启整个芯片系统,节省整个芯片系统重启浪费的时间。



1. 一种芯片,其特征在于,所述芯片包括:主处理器和多个引擎单元,每个引擎单元包括处理单元和用于控制处理单元的微控制单元;

所述主处理器,用于接收所述引擎单元中的微控制单元发送的处理单元运行信息;

响应于所述处理单元运行信息表示所述处理单元运行异常,向所述微控制单元发送恢复指令,所述恢复指令用于指示所述微控制单元恢复所述处理单元;

各个所述微控制单元,用于响应于接收到所述主处理器发送的恢复指令,恢复所述处理单元;

所述主处理器,还用于将数据包发送到所述微控制单元,所述数据包中包括待执行的业务数据;

所述微控制单元,还用于接收所述主处理器发送的数据包,控制处理单元执行所述业务数据,并将处理单元运行信息返回到所述主处理器。

2. 根据权利要求1所述的芯片,其特征在于,所述处理单元运行信息包括如下至少一项:处理单元的业务执行结果、微控制单元对处理单元的运行状况的判断结果,所述业务执行结果是所述处理单元执行业务数据后的运行结果。

3. 根据权利要求1-2任一所述的芯片,其特征在于,所述主处理器和多个引擎单元之间分别设置有传输通道;所述传输通道用于所述主处理器和所述传输通道对应的引擎单元进行数据传输。

4. 根据权利要求3所述的芯片,其特征在于,所述主处理器和所述引擎单元进行数据传输的数据包中携带有对应引擎单元的引擎单元标识。

5. 根据权利要求1所述的芯片,其特征在于,

所述主处理器,还用于加载通道驱动程序,分别生成所述主处理器和各个所述引擎单元之间的传输通道。

6. 根据权利要求5所述的芯片,其特征在于,所述芯片还包括双倍速率同步动态随机存储器DDR SDRAM,所述DDR SDRAM中预先存储有通道驱动程序。

7. 根据权利要求1-6任一所述的芯片,其特征在于,

所述主处理器,还用于接收所述微控制单元返回的所述恢复指令的恢复结果;

响应于所述恢复指令的恢复结果表示未能恢复所述处理单元,向所述微控制单元重新发送恢复指令。

8. 一种处理单元的恢复方法,其特征在于,芯片包括主处理器和多个引擎单元,每个引擎单元包括所述处理单元和用于控制处理单元的微控制单元,所述方法应用于所述主处理器,所述方法包括:

将数据包发送到所述微控制单元,所述数据包中包括待执行的业务数据;

接收所述引擎单元中的微控制单元发送的处理单元运行信息,所述处理单元运行信息由所述微控制单元控制处理单元执行所述业务数据后生成;

响应于所述处理单元运行信息表示所述处理单元运行异常,向所述微控制单元发送恢复指令,所述恢复指令用于指示所述微控制单元恢复所述处理单元。

9. 一种处理单元的恢复方法,其特征在于,芯片包括主处理器和多个引擎单元,每个引擎单元包括所述处理单元和用于控制处理单元的微控制单元,所述方法应用于所述微控制单元,所述方法包括:

接收所述主处理器发送的数据包,所述数据包中包括待执行的业务数据;
控制处理单元执行所述业务数据,并将处理单元运行信息返回到所述主处理器;
响应于接收到所述主处理器发送的恢复指令,恢复所述处理单元。

10. 一种处理单元的恢复方法,其特征在于,所述方法应用于芯片,所述芯片包括主处理器和多个引擎单元,每个引擎单元包括所述处理单元和用于控制处理单元的微控制单元;所述方法包括:

主处理器将数据包发送到所述微控制单元,所述数据包中包括待执行的业务数据;

微控制单元接收所述主处理器发送的数据包,控制处理单元执行所述业务数据,并将处理单元运行信息返回到所述主处理器;

主处理器接收所述引擎单元中的微控制单元发送的处理单元运行信息;

响应于所述处理单元运行信息表示所述处理单元运行异常,主处理器向所述微控制单元发送恢复指令,所述恢复指令用于指示所述微控制单元重启所述处理单元;

微控制单元接收所述恢复指令,并恢复所述处理单元。

11. 一种电子设备,其特征在于,所述设备包括芯片,所述芯片为权利要求1至7任一所述的芯片。

12. 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,所述程序被处理器执行时实现权利要求8至10任一所述的方法。

一种芯片和处理单元的恢复方法

技术领域

[0001] 本公开实施例涉及网络通信技术领域,尤其涉及一种芯片和处理单元的恢复方法。

背景技术

[0002] AI(Artificial Intelligence,人工智能)芯片内部一般包含主处理器和多个IP Core(Intellectual Property core,IP核),如多路AI引擎。AI业务应用程序主要运行在主处理器上,通过AI引擎驱动程序,控制相应的AI引擎进行大量算法的处理。

[0003] 由于AI算法的复杂和多样性,AI引擎在运行AI算法的过程中可能会碰到错误的执行命令,AI引擎因此进入了异常状态。由于AI引擎无法自己恢复,导致整个AI业务应用程序运行结果错误,异常退出运行,而且这一路AI引擎将无法继续使用。

[0004] 当前的解决方法一般是通过重启整个AI芯片,让内部所有的IP Core断电重启来实现对出现异常的这一路AI引擎进行恢复的目的,而这样势必对其他路的AI引擎正常运行的业务造成干扰,终止所有的业务应用程序,以及造成时间的浪费,整个SOC重启后的恢复过程也带来能源上的浪费。

发明内容

[0005] 有鉴于此,本公开实施例提供至少一种芯片和处理单元的恢复方法。

[0006] 具体地,本公开实施例是通过如下技术方案实现的:

[0007] 第一方面,提供一种芯片,所述芯片包括:主处理器和多个引擎单元,每个引擎单元包括处理单元和用于控制处理单元的微控制单元。

[0008] 所述主处理器,用于接收所述引擎单元中的微控制单元发送的处理单元运行信息。

[0009] 响应于所述处理单元运行信息表示所述处理单元运行异常,向所述微控制单元发送恢复指令,所述恢复指令用于指示所述微控制单元恢复所述处理单元。

[0010] 各个所述微控制单元,用于响应于接收到所述主处理器发送的恢复指令,恢复所述处理单元。

[0011] 第二方面,提供一种处理单元的恢复方法,所述方法应用于主处理器,芯片包括主处理器和多个引擎单元,每个引擎单元包括处理单元和用于控制处理单元的微控制单元,所述方法包括:

[0012] 将数据包发送到所述微控制单元,所述数据包中包括待执行的业务数据。

[0013] 接收所述引擎单元中的微控制单元发送的处理单元运行信息。

[0014] 响应于所述处理单元运行信息表示所述处理单元运行异常,向所述微控制单元发送恢复指令,所述恢复指令用于指示所述微控制单元重启所述处理单元。

[0015] 第三方面,提供一种处理单元的恢复方法,所述方法应用于微控制单元,芯片包括主处理器和多个引擎单元,每个引擎单元包括处理单元和用于控制处理单元的微控制单

元,所述方法包括:

[0016] 接收所述主处理器发送的数据包,所述数据包中包括待执行的业务数据。

[0017] 控制处理单元执行所述业务数据,并将处理单元运行信息返回到所述主处理器。

[0018] 响应于接收到所述主处理器发送的恢复指令,重启所述处理单元。

[0019] 第四方面,提供一种处理单元的恢复方法,所述方法应用于芯片,所述芯片包括主处理器和多个引擎单元,每个引擎单元包括处理单元和用于控制处理单元的微控制单元;所述方法包括:

[0020] 主处理器接收所述引擎单元中的微控制单元发送的处理单元运行信息。

[0021] 响应于所述处理单元运行信息表示所述处理单元运行异常,主处理器向所述微控制单元发送恢复指令,所述恢复指令用于指示所述微控制单元重启所述处理单元。

[0022] 微控制单元接收所述恢复指令,并重启所述处理单元。

[0023] 第五方面,提供一种电子设备,所述设备包括芯片,所述芯片为本公开任一实施例所述的芯片。

[0024] 第六方面,提供一种计算机可读存储介质,其上存储有计算机程序,所述程序被处理器执行时实现本公开任一实施例所述的方法。

[0025] 本公开实施例提供的芯片中的各个处理单元配置有微控制单元,由主处理器通过微控制单元对处理单元进行控制,可以对单路发生异常情况的处理单元进行自动恢复,不再需要重启整个芯片系统,以使其他路处理单元的业务流程可以继续运行,可以节省整个芯片系统重启浪费的时间。

附图说明

[0026] 为了更清楚地说明本公开一个或多个实施例或相关技术中的技术方案,下面将对实施例或相关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公开一个或多个实施例中记载的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动性的前提下,还可以根据这些附图获得其他的附图。

[0027] 图1是本公开实施例示出的一种芯片的架构;

[0028] 图2是本公开实施例示出的另一种芯片的架构;

[0029] 图3是本公开实施例示出的一种处理单元的恢复方法的流程图;

[0030] 图4是本公开实施例示出的又一种处理单元的恢复方法的流程图;

[0031] 图5是本公开实施例示出的另一种处理单元的恢复方法的流程图。

具体实施方式

[0032] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本说明书相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本说明书的一些方面相一致的方法的例子。

[0033] 在本说明书使用的术语是仅仅出于描述特定实施例的目的,而非旨在限制本说明书。在本说明书和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式,除非上下文清楚地表示其他含义。还应当理解,本文中使用的术语“和/或”是

指并包含一个或多个相关联的列出项目的任何或所有可能组合。

[0034] 应当理解,尽管在本说明书可能采用术语第一、第二、第三等来描述各种信息,但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如,在不脱离本说明书范围的情况下,第一信息也可以被称为第二信息,类似地,第二信息也可以被称为第一信息。取决于语境,如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

[0035] 如图1所示,图1示出了本公开实施例提供的一种芯片的架构,该芯片包括:主处理器和多个引擎单元,每个引擎单元包括处理单元PU(Processing Unit,处理单元)和用于控制处理单元的微控制单元MCU(MicroController Unit,微控制单元),本实施例对于引擎单元的数量不进行限制,图1以4个引擎单元为例。

[0036] 主处理器,用于接收引擎单元中的微控制单元发送的处理单元运行信息,响应于处理单元运行信息表示处理单元运行异常,向微控制单元发送恢复指令,其中,恢复指令用于指示微控制单元恢复处理单元。

[0037] 各个微控制单元,用于响应于接收到主处理器发送的恢复指令,恢复处理单元。

[0038] 本实施例中的芯片可以是各种SoC(System on Chip,系统级芯片),比如,可以是AI芯片。本实施例以AI芯片为例来进行说明。

[0039] 其中,主处理器可以是各种架构的处理器,比如,ARM(Advanced RISC Machines,高级精简指令集处理器)处理器,或者,X86架构的处理器。处理单元可以是芯片内部的IP核,IP核封装了可以实现不同功能的电路,在AI芯片中,IP核可以是AI引擎,即AI加速器或AI处理器等。业务应用程序主要运行在主处理器上,本实施例在现有芯片的基础上,为每个处理单元分别配置了微控制单元,为了便于描述,将处理单元和微控制单元的组合叫做引擎单元,由微控制单元控制处理单元与主处理器的通讯,以及控制处理单元进行相关算法的处理。

[0040] 处理单元运行信息可以是与处理单元的运行状态或运行结果相关的信息。处理单元运行信息可以由处理单元自身生成,也可以由处理单元生成。在一个例子中,处理单元运行信息可以包括如下至少一项:处理单元的业务执行结果、微控制单元对处理单元的运行状况的判断结果。业务执行结果是处理单元执行业务数据后的运行结果。

[0041] 实际实施中,主处理器可以接收四个引擎单元中任一引擎单元的微控制单元发送的处理单元运行信息,对处理单元运行信息进行判断。当判断出处理单元运行信息表示该微控制单元对应的处理单元运行异常时,可以向处理单元运行信息所属的微控制单元发送恢复指令。该恢复指令可以指示对应的微控制单元对处理单元进行恢复,以使处理单元从异常恢复为正常。

[0042] 例如,主处理器将处理单元运行信息返回给上层的业务应用程序,业务应用程序对处理单元运行信息判断,在判断出处理单元运行异常时,比如,在处理单元运行信息为业务执行结果时,判断出业务执行结果中的运行数据为未执行完成算法指令的中间数据,又比如,在处理单元运行信息为微控制单元对处理单元的运行状况的判断结果时,判断出微控制单元对处理单元的运行状况的判断结果为异常。可以通过调用预先设定的重启函数,以向微控制单元发送恢复指令。特别的,可是在判断出处理单元运行异常并且无法由处理单元自己恢复时,主处理器向微控制单元发送恢复指令。

[0043] 微控制单元在接收到恢复指令时对该处理单元单独进行恢复,恢复的方法可以是重新启动处理单元,也可以是将微控制单元和处理单元一起重新启动。

[0044] 例如,微控制单元接收到恢复指令后,可以按照该恢复指令的指示,通过设置处理单元的寄存器来重启处理单元,或者,也可以将这路的引擎单元中的处理单元和微控制单元一起重启,即除了重启处理单元以外,微控制单元自身也重启。

[0045] 本公开实施例提供的芯片中的各个处理单元配置有微控制单元,由主处理器通过微控制单元对处理单元进行控制,可以对单路发生异常情况的处理单元进行自动恢复,不再需要重启整个芯片系统,以使其他路处理单元的业务流程可以继续运行,可以节省整个芯片系统重启浪费的时间。

[0046] 在一种实施方式中,该芯片中的主处理器,还用于将数据包发送到所述微控制单元;微控制单元,还用于接收主处理器发送的数据包,控制处理单元执行业务数据,并将处理单元运行信息返回到主处理器。

[0047] 其中,数据包中包括待执行业务数据。业务数据可以是多种类型的数据,比如可以是与业务算法相关的指令流,参数以及输入数据源等。

[0048] 其中,处理单元的业务执行结果是处理单元执行业务数据后的运行结果。

[0049] 比如,在业务数据包括计算输入图片的相似度的指令和两张输入图片的存储地址时,此时,业务执行结果可以包括处理单元根据业务数据计算得到的相似度,也可以是由于中间运行错误只包含计算得到的一部分与相似度有关的数据。

[0050] 微控制单元对处理单元的运行状况的判断结果是微控制单元对处理单元运行时或运行完毕后的状态的判断结果。比如,微控制单元可以在处理单元执行业务数据时或者执行业务数据后,对处理单元的运行状况进行判断,处理单元可能因为执行了错误的算法指令而进入异常状态,微控制单元在判断出处理单元为异常状态时,将包含判断结果的业务执行结果发送到主处理器。这样在处理单元为异常状态时而无法产生任何运行数据时,由微控制单元及时发现处理单元的异常状态并返回给主处理器。

[0051] 实际实施中,主处理器可以以数据包的形式将业务应用程序所要求执行业务数据发送到微控制单元。

[0052] 微控制单元控制处理单元执行业务数据可以通过配置处理单元的寄存器来进行,比如,微控制单元将数据包中待执行业务数据写入处理单元的寄存器,以使处理单元获取AI算法指令和相应的参数并执行该指令,从而实现相应的业务算法。

[0053] 主处理器接收到微控制单元返回的处理单元运行信息后,可以将处理单元运行信息反馈给在主处理器上运行的业务应用程序,以使对主处理器根据处理单元运行信息判断处理单元是否运行异常。业务应用程序还可以从处理单元运行信息中的业务执行结果中获取业务所需要的处理单元自身产生的运行数据,运行数据可以是处理单元执行完毕相关算法指令后的运行数据,也可以是由于进入异常状态而未执行完毕相关算法指令后的中间运行数据。

[0054] 处理单元在重启后,克服原有的故障,从异常状态转为正常状态,此时,主处理器可以继续向微控制单元发送数据包,数据包可以包括重启前未执行完毕的算法指令和中间运行数据,以使微控制单元控制处理单元继续执行中断的业务流程。单路的处理单元恢复后,可以继续执行发生异常前的业务流程,省去了业务流程重新从头开始运行的步骤,更加

方便使用。

[0055] 在又一种实施方式中,如图2所示,图2是本公开实施例示出的又一种芯片的架构,该芯片中主处理器和各个引擎单元之间分别设置有传输通道,传输通道用于主处理器和传输通道对应的引擎单元进行数据传输。

[0056] 该传输通道的生成方式可以是由主处理器加载通道驱动程序,分别生成所述主处理器和各个所述引擎单元之间的传输通道。

[0057] 例如,芯片启动后,自动加载预先定义的通道驱动程序,分别生成主处理器和各个引擎单元之间的传输通道。该通道驱动程序为一个协议驱动程序,以使主处理器和各个微控制单元可以按照特定的协议进行通信。该特定的协议可以由本领域技术人员根据实际需求定义。比如,可以定义一个特定的IPC(InterProcess Communication,进程间通信)通讯协议,协议驱动程序命名为IPC driver。示例性的,该特定的IPC通讯协议可以基于ring buffer(环形缓冲器)的数据结构进行定义,以使主处理器和多个引擎单元共用的芯片中的存储器来存放通信中发送和接收的数据,实现内存共享。

[0058] 该芯片中使用DDR SDRAM(Double Data Rate SDRAM,双倍速率同步动态随机存储器)作为内部存储器,DDR SDRAM中预先存储有通道驱动程序,本领域技术人员可以根据需要选取其他类型的存储器,本实施例对具体使用何种存储器不进行限制。每个PU都有自己的编号,分别是0,1,2,3,该编号也是每个PU所在的引擎单元的引擎单元标识。DDR中预先存储通道驱动程序IPC driver,在芯片启动后加载IPC driver,分别生成主处理器和各个引擎单元之间的传输通道,即传输通道0、传输通道1、传输通道2和传输通道3。

[0059] 传输通道生成后,主处理器和引擎单元之间的数据传输,比如,主处理器向微控制单元发送数据包或恢复指令,以及微控制单元向主处理器发送处理单元运行信息,可以通过传输通道发送与接收。

[0060] 在一个例子中,主处理器和引擎单元进行数据传输的数据包中携带有对应引擎单元的引擎单元标识。引擎单元标识用于标识数据包的目的或来源引擎单元。

[0061] 实际实施中,数据包包括引擎单元标识,主处理器通过引擎单元标识确定传输通道,以使数据包通过该传输通道发送到微控制单元。比如,业务程序应用需要使用编号为2的pu进行相关算法的处理时,可以将数据包的引擎单元标识设置为2,主处理器在发送数据包时,根据数据包中的引擎单元标识,将数据包通过传输通道2发送到引擎单元标识为2的引擎单元中的微控制单元。

[0062] 本例中的数据包中的业务数据可以是业务算法所用到的数据在内存中的地址。比如,主处理器需要将一张图片发送到引擎单元,可以不把描述图片的代码发送到引擎单元,而是将该图片在芯片内存的地址通过数据包发送到引擎单元,该芯片内存是主处理器和多个PU共享的存储空间,如图2中的DDR SDRAM。PU可以根据图片的地址在DDR SDRAM中查找到描述图片的代码,而获取该图片的信息,以进行后续运算。地址的所占用的空间显然比描述图片的代码要小得多,从而使主处理器和引擎单元之间的通信更加高效快速。

[0063] 微控制单元接收到数据包后,将数据包中的业务数据写入PU的寄存器,控制PU执行业务数据,还可以对PU的运行状况进行判断,将PU自身产生的运行数据和对PU的运行状况的判断结果打包为处理单元运行信息,通过与主处理器的传输通道发送到主处理器。比如,沿用上例,微控制单元通过与主处理器的传输通道2发送业务执行结果到主处理器。

[0064] 微控制单元向主处理器返回的处理单元运行信息也可以包括对应的引擎单元标识,以使主处理器在接收到处理单元运行信息时,可以根据引擎单元标识识别出该处理单元运行信息具体来源于哪个处理单元;又比如,处理单元运行信息可以不包括对应的引擎单元标识,主处理器可以通过接收到该处理单元运行信息的传输通道判断出该处理单元运行信息具体来源于哪个处理单元,为该处理单元运行信息打上对应的引擎单元标识,以便于后续步骤中主处理器和微控制单元的数据通信。

[0065] 需要说明的是,PU执行相关算法指令后可以将得到的运行数据存入芯片内存,业务执行结果同样也可以是运行数据在芯片内存的地址,比如,PU可以是对某个图像进行了场景分割处理,得到场景分割图,将描述该场景分割图的代码存入芯片内存,业务执行结果携带该代码的地址,而不用携带完整描述该场景分割图的代码。

[0066] 主处理器将接收到的处理单元运行信息反馈给业务应用程序,业务应用程序可以从处理单元运行信息中获取所需要的处理单元执行算法指令后的运行数据,并判断处理单元是否运行异常。

[0067] 当处理单元运行信息表示处理单元运行异常,主处理器可以调用驱动运行程序中预先设定的重启函数,以向微控制单元发送恢复指令。恢复指令也可以包括引擎单元标识,主处理器向微控制单元发送恢复指令时,通过引擎单元标识对应的传输通道来发送。

[0068] 微控制单元通过和主处理器之间的传输通道接收恢复指令,按照该恢复指令的指示,通过设置处理单元的寄存器来重启处理单元,或者,重启该路引擎单元。

[0069] 在另一种实施方式中,主处理器,还用于接收微控制单元返回的恢复指令的恢复结果;响应于恢复指令的恢复结果表示未能恢复处理单元,向微控制单元重新发送恢复指令。

[0070] 微控制单元重启处理单元后,可以将处理单元的恢复结果,即重启后的结果返回到主处理器。若处理单元重启后恢复到正常状态,则恢复指令的恢复结果为成功恢复处理单元;若处理单元重启后仍为异常状态,则恢复指令的恢复结果为未能恢复处理单元。

[0071] 当恢复指令的恢复结果为成功恢复处理单元时,主处理器可以继续向微控制单元发送数据包,数据包可以包括重启前未执行完毕的算法指令和中间运行数据,以使微控制单元控制处理单元继续执行中断的业务流程。

[0072] 当恢复指令的恢复结果为未能恢复处理单元时,主处理器可以向微控制单元重新发送恢复指令,以尝试重新恢复处理单元。当向某个微控制单元发送的恢复指令达到预设阈值时,主处理器可以重启整个芯片。

[0073] 本例在单次恢复处理单元失败后,可以尝试重新恢复,增大了恢复成功的概率;单路的处理单元恢复后,可以继续执行发生异常前的业务流程,省去了业务流程重新从头开始运行的步骤,更加方便使用。

[0074] 本公开实施例还提供了一种处理单元的恢复方法,该方法应用于芯片中的主处理器,芯片包括主处理器和多个引擎单元,每个引擎单元包括处理单元和用于控制处理单元的微控制单元,如图3所示,该方法包括以下步骤,主处理器的具体执行动作在上述实施例中已进行详细说明,这里不再赘述:

[0075] 在步骤302中,将数据包发送到所述微控制单元,所述数据包中包括待执行的业务数据。

[0076] 在一个例子中,主处理器和各个引擎单元之间分别设置有传输通道;所述传输通道用于所述主处理器和所述传输通道对应的引擎单元进行数据传输。

[0077] 本步骤之前,主处理器可以加载通道驱动程序,分别生成主处理器和各个引擎单元之间的传输通道。

[0078] 在一个例子中,主处理器和引擎单元进行数据传输的数据包中携带有对应引擎单元的引擎单元标识。

[0079] 在一个例子中,芯片还包括DDR SDRAM,DDR SDRAM中预先存储有通道驱动程序。

[0080] 在步骤304中,接收引擎单元中的微控制单元发送的处理单元运行信息。

[0081] 在一个例子中,可以通过生成的传输通道,主处理器接收引擎单元上的微控制单元发送的业务执行结果。

[0082] 在一个例子中,处理单元运行信息包括如下至少一项:处理单元的业务执行结果、微控制单元对处理单元的运行状况的判断结果,业务执行结果是所述处理单元执行业务数据后的运行结果。

[0083] 在步骤306中,响应于处理单元运行信息表示处理单元运行异常,向微控制单元发送恢复指令,所述恢复指令用于指示所述微控制单元恢复处理单元。

[0084] 在一个例子中,可以通过引擎单元标识对应的传输通道,主处理器向微控制单元发送恢复指令。

[0085] 在一个例子中,主处理器可以接收微控制单元返回的恢复指令的恢复结果,响应于恢复指令的恢复结果表示未能恢复处理单元,向微控制单元重新发送恢复指令。

[0086] 本公开实施例提供的处理单元的恢复方法,为芯片内的各个处理单元配置微控制单元,由微控制单元对处理单元进行控制,可以对单路发生异常情况的处理单元进行自动恢复,不再需要重启整个芯片系统,以使其他路处理单元的业务流程可以继续运行,可以节省整个芯片系统重启浪费的时间。

[0087] 本公开实施例还提供了一种芯片内部处理单元的恢复方法,该方法应用于芯片中的微控制单元,芯片包括主处理器和多个引擎单元,每个引擎单元包括处理单元和用于控制处理单元的微控制单元,如图4所示,该方法包括以下步骤,微控制单元的具体执行动作在上述实施例中已进行详细说明,这里不再赘述:

[0088] 在步骤402中,接收所述主处理器发送的数据包,所述数据包中包括待执行业务数据。

[0089] 在一个例子中,主处理器和各个引擎单元之间分别设置有传输通道;所述传输通道用于所述主处理器和所述传输通道对应的引擎单元进行数据传输。微控制单元可以通过主处理器和引擎单元之间的传输通道,接收主处理器发送的数据包。

[0090] 在一个例子中,主处理器和所述引擎单元进行数据传输的数据包中携带有对应引擎单元的引擎单元标识。

[0091] 在步骤404中,控制处理单元执行所述业务数据,并将处理单元运行信息返回到所述主处理器。

[0092] 在一个例子中,处理单元运行信息包括如下至少一项:处理单元的业务执行结果、微控制单元对处理单元的运行状况的判断结果,所述业务执行结果是所述处理单元执行业务数据后的运行结果。

[0093] 在步骤406中,响应于接收到所述主处理器发送的恢复指令,恢复所述处理单元。

[0094] 在一个例子中,微控制单元可以通过主处理器和引擎单元之间的传输通道,接收主处理器发送的恢复指令。

[0095] 本公开实施例提供的处理单元的恢复方法,为芯片内的各个处理单元配置微控制单元,由微控制单元对处理单元进行控制,可以对单路发生异常情况的处理单元进行自动恢复,不再需要重启整个芯片系统,以使其他路处理单元的业务流程可以继续运行,可以节省整个芯片系统重启浪费的时间。

[0096] 本公开实施例提供了一种处理单元的恢复方法,该方法应用于芯片,芯片包括主处理器和多个引擎单元,每个引擎单元包括处理单元和用于控制处理单元的微控制单元,如图5所示,该方法包括以下步骤,上述主处理器和微控制单元的功能参见上述实施例,这里不再赘述。

[0097] 在步骤502中,主处理器接收所述引擎单元中的微控制单元发送的处理单元运行信息。

[0098] 在一个例子中,处理单元运行信息包括如下至少一项:处理单元的业务执行结果、微控制单元对处理单元的运行状况的判断结果,所述业务执行结果是所述处理单元执行业务数据后的运行结果。

[0099] 在一个例子中,在本步骤之前,主处理器还用于加载通道驱动程序,分别生成所述主处理器和各个所述引擎单元之间的传输通道。

[0100] 在一个例子中,芯片还包括DDR SDRAM,所述DDR SDRAM中预先存储有通道驱动程序。

[0101] 在一个例子中,主处理器和各个引擎单元之间分别设置有传输通道;所述传输通道用于所述主处理器和所述传输通道对应的引擎单元进行数据传输。

[0102] 在一个例子中,在本步骤之前,主处理器,还用于将数据包发送到所述微控制单元,所述数据包中包括待执行业务数据;微控制单元,还用于接收所述主处理器发送的数据包,控制处理单元执行所述业务数据,并将处理单元运行信息返回到所述主处理器。

[0103] 在一个例子中,主处理器和所述引擎单元进行数据传输的数据包中携带有对应引擎单元的引擎单元标识。

[0104] 在步骤504中,响应于所述处理单元运行信息表示所述处理单元运行异常,主处理器向所述微控制单元发送恢复指令,所述恢复指令用于指示所述微控制单元重启所述处理单元。

[0105] 在步骤506中,微控制单元接收所述恢复指令,并恢复所述处理单元。

[0106] 在一个例子中,主处理器还用于接收所述微控制单元返回的所述恢复指令的恢复结果;响应于所述恢复指令的恢复结果表示未能恢复所述处理单元,向所述微控制单元重新发送恢复指令。

[0107] 本公开实施例提供的处理单元的恢复方法,为芯片内的各个处理单元配置微控制单元,由微控制单元对处理单元进行控制,可以对单路发生异常情况的处理单元进行自动恢复,不再需要重启整个芯片系统,以使其他路处理单元的业务流程可以继续运行,可以节省整个芯片系统重启浪费的时间。

[0108] 本公开实施例还提供了一种电子设备,该电子设备包括芯片,所述芯片为本公开

任一实施例所述的芯片。

[0109] 本公开实施例还提供了一种计算机可读存储介质,其上存储有计算机程序,所述程序被处理器执行时实现本公开任一实施例所述的处理单元的恢复方法。

[0110] 对于设备实施例而言,由于其基本对应于方法实施例,所以相关之处参见方法实施例的部分说明即可。以上所描述的设备实施例仅仅是示意性的,其中所述作为分离部件说明的模块可以是或者也可以不是物理上分开的,作为模块显示的部件可以是或者也可以不是物理模块,即可以位于一个地方,或者也可以分布到多个网络模块上。可以根据实际的需要选择其中的部分或者全部模块来实现本说明书方案的目的。本领域普通技术人员在不付出创造性劳动的情况下,即可以理解并实施。

[0111] 上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下,在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外,在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中,多任务处理和并行处理也是可以的或者可能是有利的。

[0112] 本领域技术人员在考虑说明书及实践这里申请的发明后,将容易想到本说明书的其它实施方案。本说明书旨在涵盖本说明书的任何变型、用途或者适应性变化,这些变型、用途或者适应性变化遵循本说明书的一般性原理并包括本说明书未申请的本技术领域中的公知常识或惯用技术手段。说明书和实施例仅被视为示例性的,本说明书的真正范围和精由下面的权利要求指出。

[0113] 应当理解的是,本说明书并不局限于上面已经描述并在附图中示出的精确结构,并且可以在不脱离其范围进行各种修改和改变。本说明书的范围仅由所附的权利要求来限制。

[0114] 以上所述仅为本说明书的较佳实施例而已,并不用以限制本说明书,凡在本说明书的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本说明书保护的范围内。

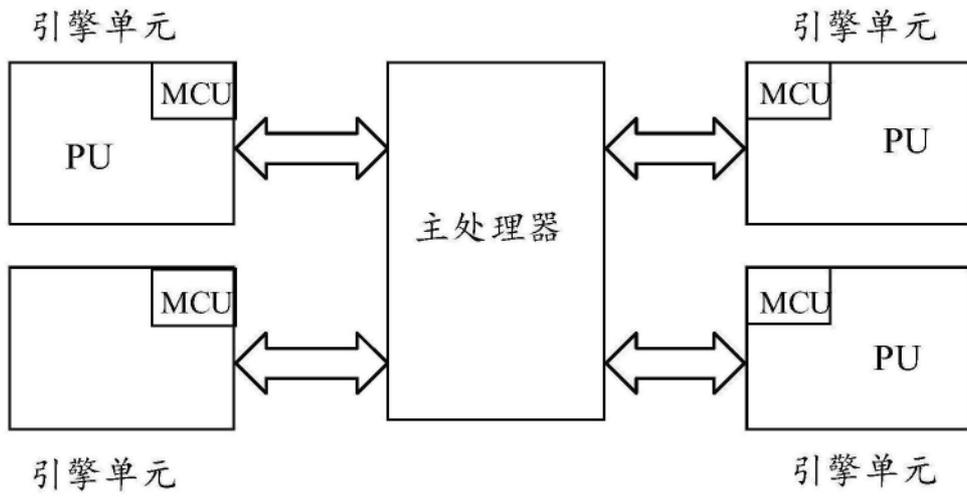


图1

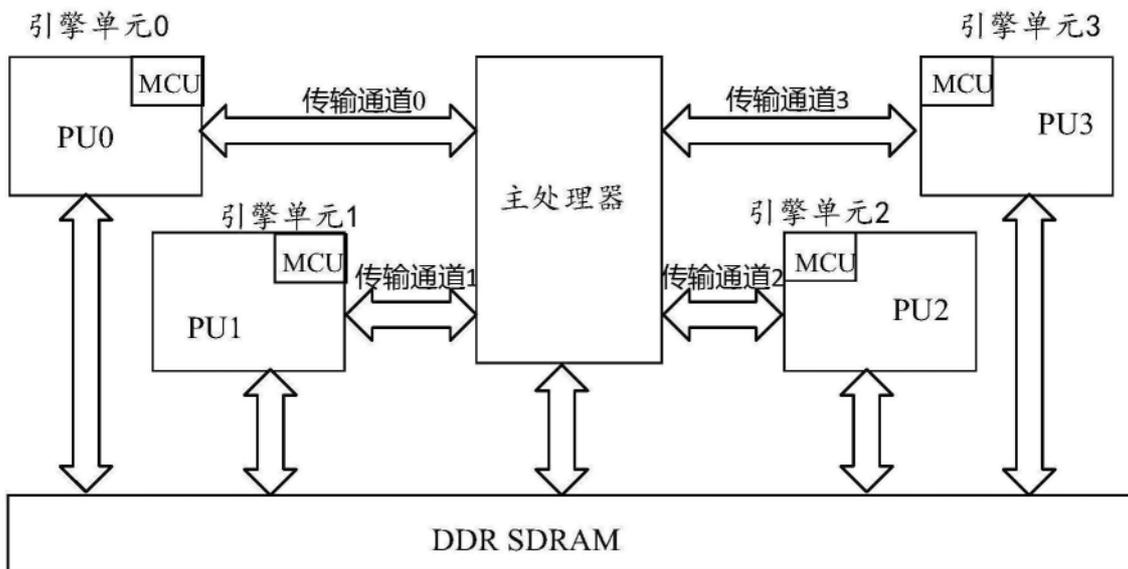


图2

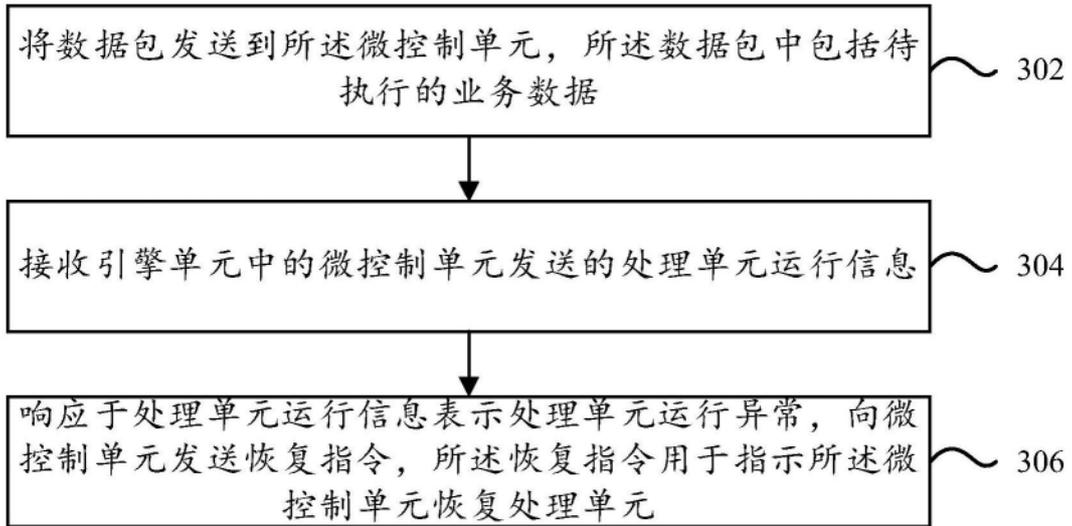


图3

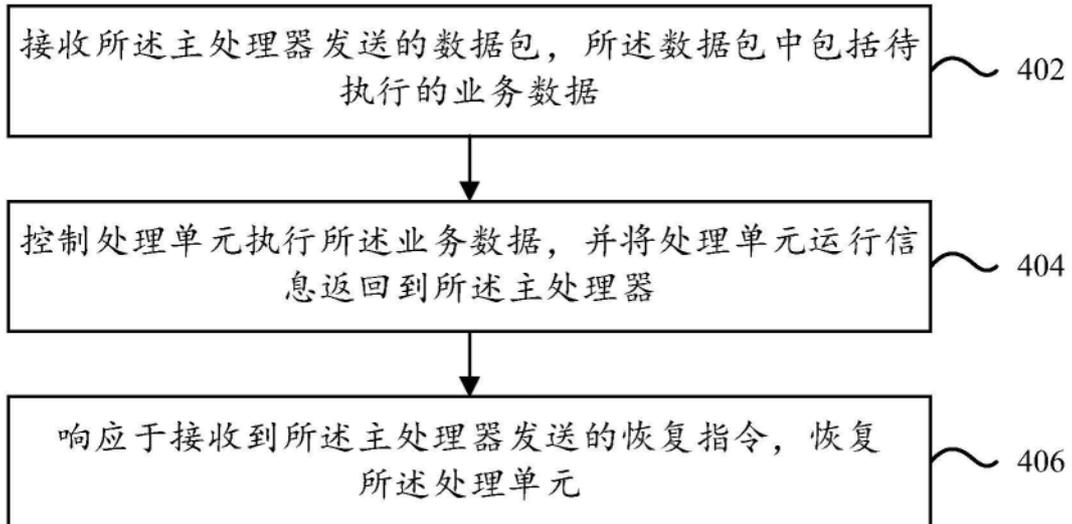


图4

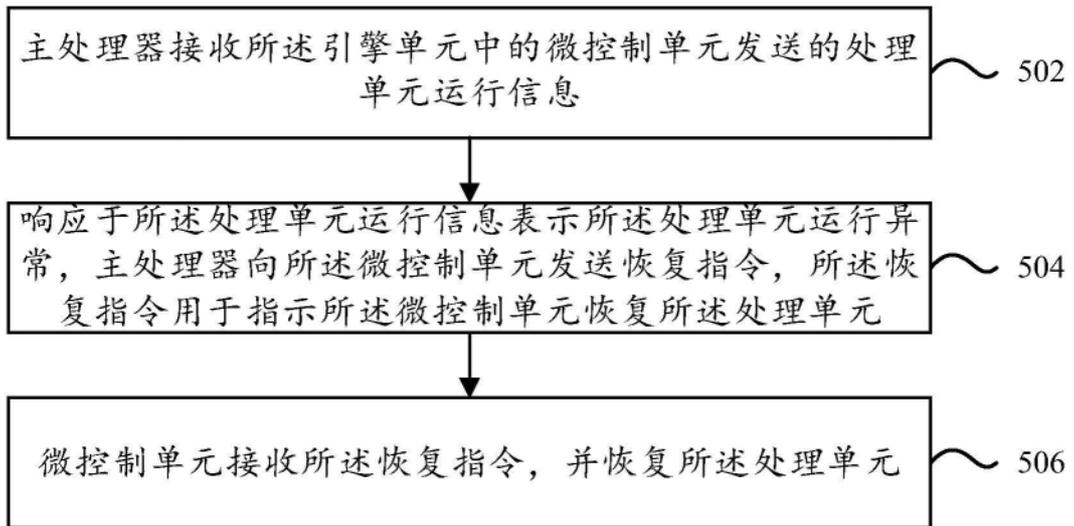


图5