



(12) 发明专利申请

(10) 申请公布号 CN 103186359 A

(43) 申请公布日 2013. 07. 03

(21) 申请号 201110454003. 9

(22) 申请日 2011. 12. 30

(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 元西西 毛二坤 汪坚 田学红
曾代兵 田万廷 陈前

(74) 专利代理机构 北京派特恩知识产权代理事
务所(普通合伙) 11270

代理人 王黎延 周义刚

(51) Int. Cl.

G06F 5/06 (2006. 01)

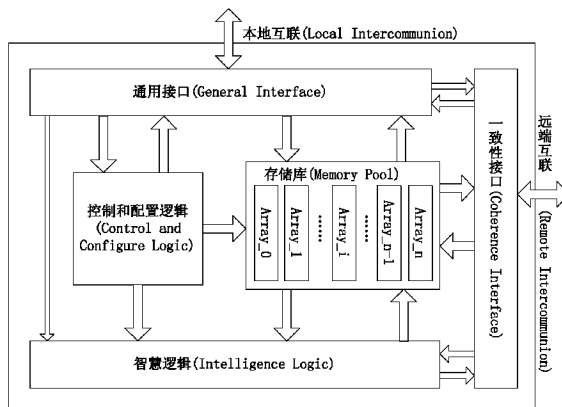
权利要求书1页 说明书6页 附图7页

(54) 发明名称

硬件抽象数据结构、数据处理方法及系统

(57) 摘要

本发明公开了一种硬件抽象数据结构,包括:通用接口(GI)、一致性接口(CI)、控制和配置逻辑(CCL)、智慧逻辑(IL)和存储库(MP);其中,所述GI,用于实现硬件抽象数据结构(HADS)与处理器之间的交互;所述CI,用于实现多处理器间的一致性存储;所述CCL,用于根据GI接收的配置命令对MP进行硬件数据结构配置;所述IL,用于完成简单、大量、频繁的数据处理;所述MP,用于存储数据。本发明还相应地公开了一种数据处理方法及系统。通过本发明,能够实现可动态配置、灵活多变、高效率、接口通用、互联特性好的硬件抽象数据结构,进而提高数据处理效率。



1. 一种硬件抽象数据结构,其特征在于,该硬件抽象数据结构HADS包括:通用接口GI、控制和配置逻辑CCL和存储库MP;其中,
所述GI,用于实现HADS与处理器之间的交互;
所述CCL,用于根据GI接收的配置命令对MP进行硬件数据结构配置;
所述MP,用于存储数据。
2. 根据权利要求1所述的硬件抽象数据结构,其特征在于,该HADS还包括一致性接口CI,
所述CI,用于多核之间的互联以及直接数据访问。
3. 根据权利要求1或2所述的硬件抽象数据结构,其特征在于,该HADS还包括智慧逻辑IL,
所述CCL,还用于根据GI接收的命令控制数据在MP到接口之间的传输、数据在IL内的处理、以及数据的旁路;
所述IL,用于根据CCL的控制,对MP中存储的数据进行处理。
4. 根据权利要求3所述的硬件抽象数据结构,其特征在于,
所述IL,还用于根据CCL的控制输出处理后的数据,或将处理后的数据写回MP。
5. 一种数据处理方法,其特征在于,该方法包括:
处理器对HADS进行配置,将HADS内的存储块配置成需要的硬件数据结构,之后采用所述配置的存储块进行数据存储。
6. 根据权利要求5所述的数据处理方法,其特征在于,所述处理器对HADS进行配置,将HADS内的存储块配置成需要的硬件数据结构为:
处理器通过GI将配置命令发送给CCL,所述配置命令至少携带需要配置的硬件数据结构信息;
所述CCL根据所述配置命令对MP进行硬件数据结构配置,将MP中的存储块配置成需要的硬件数据结构。
7. 根据权利要求6所述的数据处理方法,其特征在于,该方法还包括:
处理器通过CI或GI直接访问MP内的数据;
所述CCL控制数据在MP与接口之间的传输、数据在IL内的处理、以及数据在CI与GI间的旁路。
8. 根据权利要求7所述的数据处理方法,其特征在于,该方法还包括:
所述CCL控制IL输出处理后的数据,或将处理后的数据写回MP。
9. 根据权利要求5至8任一项所述的数据处理方法,其特征在于,所述处理器为本地处理器和/或远端处理器。
10. 一种数据处理系统,其特征在于,该系统包括处理器和权利要求1至4任一项所述的HADS。
11. 根据权利要求10所述的数据处理系统,其特征在于,所述处理器为本地处理器和/或远端处理器。

硬件抽象数据结构、数据处理方法及系统

技术领域

[0001] 本发明涉及数据存储技术,尤其涉及一种硬件抽象数据结构(Hardware Abstract Data Structure, HADS)、数据处理方法及系统。

背景技术

[0002] 用处理器处理一个具体任务时,需要将具体任务抽象成适当的数据结构,基本的类型包括集合、线性结构、树形结构、图状结构等,这些结构的实现大都是通过软件实现的,效率低、实现复杂、能耗比较大。下面以堆栈(STACK)为例说明软件如何实现特定的数据结构。

[0003] STACK是限定仅在栈顶(TOP)和栈底(BASE)进行插入或删除操作的线性表。实现软件STACK的过程是先根据栈的最大使用量申请存储空间,设置STACK栈底指针和栈顶指针,初始化时STACK为NULL,栈顶指针指向栈底,即 $TOP = BASE$,然后,通过压入(PUSH)指令将数据压入STACK,每执行一次PUSH操作, TOP加1;通过弹出(POP)指令弹出STACK保存的数据,每执行一次POP操作, TOP减1。这里的TOP和BASE指针都是保存在寄存器或存储器内的,加1或减1操作也需要通过指令来完成。软件STACK的实现原理如图1所示,可以看出,对STACK的每次操作至少需要两条指令,如果需要通过实现STACK的防覆盖等功能,每次操作将需要更多的指令,从而STACK的处理效率较低。

[0004] 虽然目前出现了硬件实现的数据结构,但是也只能实现结构较简单的数据结构,如先入先出队列(FIFO)、STACK、堆(HEAP)、队列(QUEUE)等集合或线性结构,但由于现有硬件实现的数据结构通常是采用固定硬件实现的,从而可配置性不强、不灵活、浪费系统资源。硬件实现的FIFO结构如图2所示,可以看出,硬件FIFO由读地址生成(Read Address Generate)、写地址生成(Write Address Generate)、双端口SRAM(Double Port RAM)、FIFO的空满状态(Full and Empty Flag)组成,同时还需要考虑输入输出接口的兼容性问题。虽然硬件实现的FIFO效率相当高,设置不需要配置就可以直接使用,但是实现的FIFO大小、接口等都是固定不变的,另外,当处理器不用FIFO的时候,这部分资源白白浪费了,不可能将这种硬件FIFO用作HEAP或其它结构。如果采用硬件实现TREE结构,除了不能重用外,设计也较复杂。

发明内容

[0005] 有鉴于此,本发明的主要目的在于提供一种硬件抽象数据结构、数据处理方法及系统,能够实现可动态配置、灵活多变、高效率、接口通用、互联特性好的硬件抽象数据结构,提高数据处理效率。

[0006] 为达到上述目的,本发明的技术方案是这样实现的:

[0007] 一种硬件抽象数据结构,该硬件抽象数据结构HADS包括:通用接口GI、控制和配置逻辑CCL和存储库MP;其中,

[0008] 所述GI,用于实现HADS与处理器之间的交互;

- [0009] 所述 CCL,用于根据 GI 接收的配置命令对 MP 进行硬件数据结构配置;
- [0010] 所述 MP,用于存储数据。
- [0011] 该 HADS 还包括一致性接口 CI,
- [0012] 所述 CI,用于多核之间的互联以及直接数据访问。
- [0013] 该 HADS 还包括智慧逻辑 IL,
- [0014] 所述 CCL,还用于根据 GI 接收的命令控制数据在 MP 到接口之间的传输、数据在 IL 内的处理、以及数据的旁路;
- [0015] 所述 IL,用于根据 CCL 的控制,对 MP 中存储的数据进行处理。
- [0016] 所述 IL,还用于根据 CCL 的控制输出处理后的数据,或将处理后的数据写回 MP。
- [0017] 一种数据处理方法,包括:
- [0018] 处理器对 HADS 进行配置,将 HADS 内的存储块配置成需要的硬件数据结构,之后采用所述配置的存储块进行数据存储。
- [0019] 所述处理器对 HADS 进行配置,将 HADS 内的存储块配置成需要的硬件数据结构为:
- [0020] 处理器通过 GI 将配置命令发送给 CCL,所述配置命令至少携带需要配置的硬件数据结构信息;
- [0021] 所述 CCL 根据所述配置命令对 MP 进行硬件数据结构配置,将 MP 中的存储块配置成需要的硬件数据结构。
- [0022] 该方法还包括:
- [0023] 处理器通过 CI 或 GI 直接访问 MP 内的数据;
- [0024] 所述 CCL 控制数据在 MP 与接口之间的传输、数据在 IL 内的处理、以及数据在 CI 与 GI 间的旁路。
- [0025] 该方法还包括:
- [0026] 所述 CCL 控制 IL 输出处理后的数据,或将处理后的数据写回 MP。
- [0027] 所述处理器为本地处理器和 / 或远端处理器。
- [0028] 一种数据处理系统,包括处理器和权利要求 1 至 4 任一项所述的 HADS。
- [0029] 所述处理器为本地处理器和 / 或远端处理器。
- [0030] 本发明硬件抽象数据结构、数据处理方法及系统中的硬件抽象数据结构 (HADS) 包括:通用接口 (GI)、控制和配置逻辑 (CCL) 和存储库 (MP);其中,所述 GI,用于实现 HADS 与处理器之间的互联;所述 CCL,用于根据 GI 接收的配置命令对 MP 进行硬件数据结构配置;所述 MP,用于存储数据。本发明实现了完全可动态配置、灵活多变、高效率、接口通用、互联特性好的硬件抽象数据结构,采用所述硬件抽象数据结构来实现 Cache、TCM、CAM 等存储结构,以及实现 FIFO、BUFFER、HEAP、STACK、QUEUE、LINK、TABLE 等线性结构,以及实现 TREE 型结构以及实现图及其它结构,可以使这些结构的实现更加高效、灵活;同时,本发明具有对数据的智慧处理和转换功能,尽量将数据的处理靠近存储器、减少对总线的依赖,从而能够减轻处理器的负担,提高性能、降低功耗、节省成本。

附图说明

- [0031] 图 1 为现有软件 STACK 的实现原理示意图;

- [0032] 图 2 为现有硬件实现的 FIFO 结构示意图；
- [0033] 图 3 为本发明硬件抽象数据结构的示意图；
- [0034] 图 4 为 HADS 可实现的数据结构示意图；
- [0035] 图 5 为本发明实施例 1 中 HADS 实现 FIFO 的示意图；
- [0036] 图 6 为本发明实施例 2 中 HADS 实现 TCM 的示意图；
- [0037] 图 7 为本发明实施例 3 中 HADS 实现 STACK 的示意图；
- [0038] 图 8 为本发明实施例 4 中 HADS 实现 Cache 的示意图；
- [0039] 图 9 为本发明实施例 5 中 HADS 实现查找表的示意图；
- [0040] 图 10 为本发明实施例 6 中 HADS 实现 TREE 的示意图；
- [0041] 图 11 为 HADS 内对数据处理的实例示意图；
- [0042] 图 12 为 HADS 在系统中可能的使用情况示意图；
- [0043] 图 13 为 HADS 与其他存储器共用的实例示意图。

具体实施方式

[0044] 本发明的基本思想是：硬件抽象数据结构 (HADS) 包括：通用接口 (GI)、控制和配置逻辑 (CCL) 和存储库 (MP)；其中，所述 GI，用于实现 HADS 与处理器之间的交互；所述 CCL，用于根据 GI 接收的配置命令对 MP 进行硬件数据结构配置；所述 MP，用于存储数据。

[0045] 图 3 为本发明硬件抽象数据结构的示意图，如图 3 所示，该硬件抽象数据结构包括：通用接口 (General Interface, GI)、控制和配置逻辑 (Control and Configure Logic, CCL) 和存储库 (Memory Pool, MP)，其中，

[0046] 所述 GI，与处理器实现互联，用于实现 HADS 与处理器之间的交互；

[0047] 所述 CCL，用于根据 GI 接收的配置命令对 MP 进行硬件数据结构配置；具体的，可以将 MP 内的存储阵列 (Memory Array, MA) 动态配置成需要的硬件数据结构，如 Cache、TCM、CAM、FIFO、BUFFER、HEAP、STACK、QUEUE、TABLE、LINK、TREE、GRAPHIC 等结构。

[0048] 所述 MP，用于存储数据。这里所述的数据是一个广义的概念，除了具体的数据，还可以包括索引、指针等。MP 一般由大量的存储块（可以以阵列表示，但不限于阵列）组成，一定大小的存储器可以配置成 n 块存储阵列 (Memory Array, MA)，每个存储阵列的作用完全可配置，可能存储的是索引，也可能是指针或数据。本发明将 MP 以外的部分称为硬件抽象的数据结构智慧控制和互联 (HADS_ICI)。

[0049] 该 HADS 还包括一致性接口 (Coherence Interface, CI)，用于多核之间的互联以及直接数据访问。

[0050] 该 HADS 还包括智慧逻辑 (Intelligence Logic, IL)，

[0051] 所述 CCL，还用于根据 GI 接收的命令控制数据在 MP 到接口之间的传输、数据在 IL 内的处理、以及数据的旁路；

[0052] 所述 IL，用于根据 CCL 的控制，对 MP 中存储的数据进行处理。

[0053] 所述 IL，还用于根据 CCL 的控制输出处理后的数据，或将处理后的数据写回 MP。

[0054] HADS 还可以包括一致性接口 (Coherence Interface, CI)：用于多核之间的互联，也可用于直接数据访问（如 DMA）；

[0055] 需要说明的是，基于 HADS，可以根据软件的需求，将整个 MP 配置成不同的数据结

构,图 4 为 HADS 可实现的数据结构示意图,如图 4 所示,系统可以将 MP 配置成不同的数据结构,即使这种数据结构正在使用也可以动态的改变其属性,另外,所有这些数据输入或输出的数据或索引等都可以通过 IL 进行处理,不是所有访问都必须返回数值给处理器,数据甚至可以在不同的数据结构之间进行流动,以达到节省总线带宽,减轻处理器负担的目的,处理器仅负责复杂的运算和控制,从而提高处理性能。

[0056] 本发明还相应地提出一种数据处理方法,该方法包括:

[0057] 处理器对 HADS 进行配置,将 HADS 内的存储块配置成需要的硬件数据结构,之后采用所述配置的存储块进行数据存储。

[0058] 所述处理器对 HADS 进行配置,将 HADS 内的存储块配置成需要的硬件数据结构为:

[0059] 处理器通过 GI 将配置命令发送给 CCL,所述配置命令至少携带需要配置的硬件数据结构信息;

[0060] 所述 CCL 根据所述配置命令对 MP 进行硬件数据结构配置,将 MP 中的存储块配置成需要的硬件数据结构。

[0061] 该方法还包括:

[0062] 处理器通过 CI 或 GI 直接访问 MP 内的数据;

[0063] 所述 CCL 控制数据在 MP 到接口之间的传输、数据在 IL 内的处理、以及数据在 CI 和 GI 间的直通旁路。

[0064] 该方法还包括:

[0065] 所述 CCL 控制 IL 输出处理后的数据,或将处理后的数据写回 MP。

[0066] 所述处理器为本地处理器和 / 或远端处理器。

[0067] 本发明还相应地提出一种数据处理系统,该系统包括处理器和上述的 HADS。

[0068] 所述处理器为本地处理器和 / 或远端处理器,多个处理器间可以共享同一种或多种数据结构。

[0069] 下面结合具体实施例对本发明的技术方案作进一步详细说明。

[0070] 实施例 1

[0071] 图 5 为本发明实施例 1 中 HADS 实现 FIFO 的示意图,如图 5 所示,本实施例中,根据一条简单的在 MP 内配置一个 FIFO 的命令 (①),HADS 在 CCL 的控制 (②) 下将生成一个硬件 FIFO,这个硬件 FIFO 的特性,包括大小、数据输入输出的方向 (来自 GI 或 CI,也可以指定从 CI 或 GI 输出)、以及可以使能旁路、是否支持数据覆盖,甚至可以根据 FIFO 的状态产生中断,在使用过程中也可以使用命令适当改变 FIFO 的属性。在 IL 的支持下,可以对数据的输入 (③) 和输出 (④) 作运算或变换等操作,FIFO 的数据输入既可以来自处理器,也可以是 HADS 内其它结构输出的数据;同样输出可以直接返回给处理器,也可以送到其它的数据结构。

[0072] 实施例 2

[0073] 图 6 为本发明实施例 2 中 HADS 实现 TCM 的示意图,可以看出,通过配置,HADS 可以简单地当作 TCM 来使用,该 TCM 甚至可以在一个时钟周期返回处理器或其它数据结构需要的数据,节省对存储器的访问延迟,提高处理器性能。

[0074] 实施例 3

[0075] 图 7 为本发明实施例 3 中 HADS 实现 STACK 的示意图,图 7 所示的 STACK 完全可以达到图 1 所示 STACK 的灵活程度,当软件觉得大小不够使用时,随时可以通过指令动态配置,与图 1 不同的是,这种方法实现的 STACK 读写地址的生成完全由 IL 来做,不需要专门的地址生成逻辑。

[0076] 实施例 4

[0077] 图 8 为本发明实施例 4 中 HADS 实现 Cache 的示意图,Cache 的实现至少需要两块存储阵列,分别作为 TAG 和 DATA 使用。通过 IL 和 CCL,可以实现虚实地址转换,甚至可以使用一块存储阵列来实现 TLB 的功能。通过一致性接口,可以进行多核的互联以及实现 cache 的一致性功能。

[0078] 实施例 5

[0079] 本实施例以查找表为例说明 TABLE 的实现,图 9 为本发明实施例 5 中 HADS 实现查找表的示意图,具体的,可以使用一个存储阵列来实现查找表,也可以使用多个存储阵列。图 9 中④表示索引,可以直接查表得到数据⑥,实现一级查找表的功能,也可能查表得到的是下一个索引⑤,实现多级链表。

[0080] 实施例 6

[0081] 图 10 为本发明实施例 6 中 HADS 实现 TREE 的示意图,其中,④表示树的根 (root) 指针,⑤表示由 root 指针得到的两个分支指针,其中一个指针取到了数据⑥,到达了树的叶 (leaf),该分支结束;另一个分支指针取到的还是指针⑦,将继续追踪 (trace),直到所追踪的树叶,取得需要的数据⑧。根据软件的需要,HADS 可以实现复杂的树形结构,而限于图 10 的示例。

[0082] 需要说明的是,从图 5 到图 10 仅是列举了 HADS 可以实现的几种数据结构,在 CCL 和 IL 的控制下,HADS 可以实现所有可以想到的数据结构,数据可以在这些数据结构之间流动,IL 可以对这些数据进行处理。举例来说,两个核之间通过 FIFO 进行通信,如果两个核的大小端是动态可配置的,在它们之间传输的数据也必须根据两个核配置的变化进行大小端的匹配,如果两个核的大小端是一致的,则数据直接传输到 CI 接口;如果大小端不一致,则需要将数据按字节扭一下,通过 IL 很容易实现这个功能,HADS 内对数据处理的实例如图 11 所示。

[0083] 需要说明的是,对于矩阵变换或位操作(如位提取或插入)等数据量大的操作,使用 HADS 更方便,可以在 HADS 内定义一个 BUFFER,将处理的中间结果先存入 BUFFER 内,待数据处理完后返回给处理器。

[0084] HADS 在系统中可能的使用情况如图 12 所示,可以看出,HADS 的使用非常灵活,可以作为单核的内部存储器,也可以作为多核的共享存储器;还可以每个核拥有一个或多个 HADS,核间通过一致性接口保证多核的 HADS 内数据的一致性;当然,也可以作为外部存储器使用。

[0085] 另外,多个 HADS 能相互连接组合成较大的 HADS;HADS 与其它类型的存储器也是可以共存的,且 HADS 与其它存储器间可以相互通信。也就是说,HADS 可以适用于任何处理器架构,可以使用作为处理器的唯一存储器,也可以与其它存储器一起使用,HADS 与其他存储器共用的实例如图 13 所示。

[0086] 总的来说,与现有技术相比,HADS 具有如下特点:

[0087] 1) 动态可配置, 灵活多变, Memory 作为何种数据结构使用完全由软件决定;

[0088] 2) 效率高, 完全可以达到固定硬件实现的效率;

[0089] 3) Memory 内多种数据结构可以并存, 数据可以在各种数据结构之间流动, 减少了数据总线的流量, 有利于性能的提升和功耗的降低;

[0090] 4) 可以对各种数据结构的输入输出数据进行处理, 减轻了处理器的负担, 可以提高处理器的性能;

[0091] 5) 使用灵活, 可以适用于任何处理器架构。

[0092] 采用本发明所述的 HADS, 与现有技术相比, 取得了软件实现数据结构的灵活, 达到了硬件实现数据结构的高效, HADS 可以在任何处理器架构下使用, 完全动态可配置, 智慧逻辑具有数据处理和转换能力, 降低了总线访问的频率, 有利于性能的提高和功耗的降低。在 HADS 内, 多种数据结构可以并存, 数据可以在各种数据结构之间流动, 智慧逻辑可以对流动的数据进行处理, 减轻总线和处理器的负担, 从而达到提升性能和降低功耗的目的。

[0093] 以上所述, 仅为本发明的较佳实施例而已, 并非用于限定本发明的保护范围。

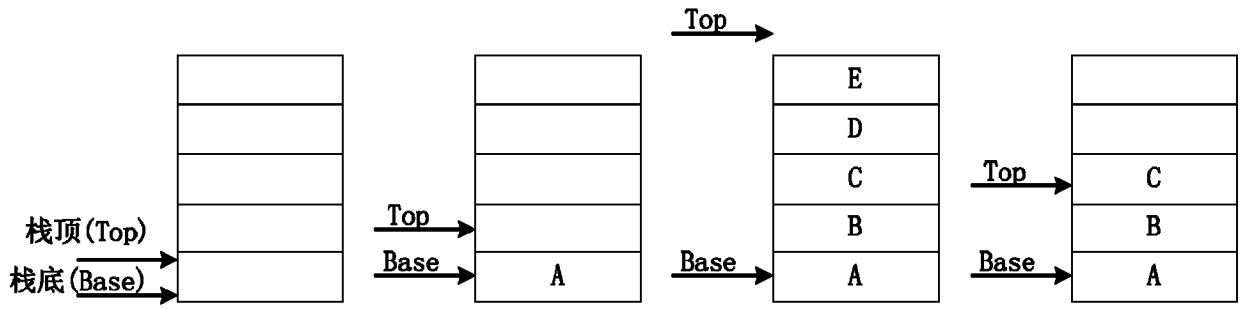


图 1

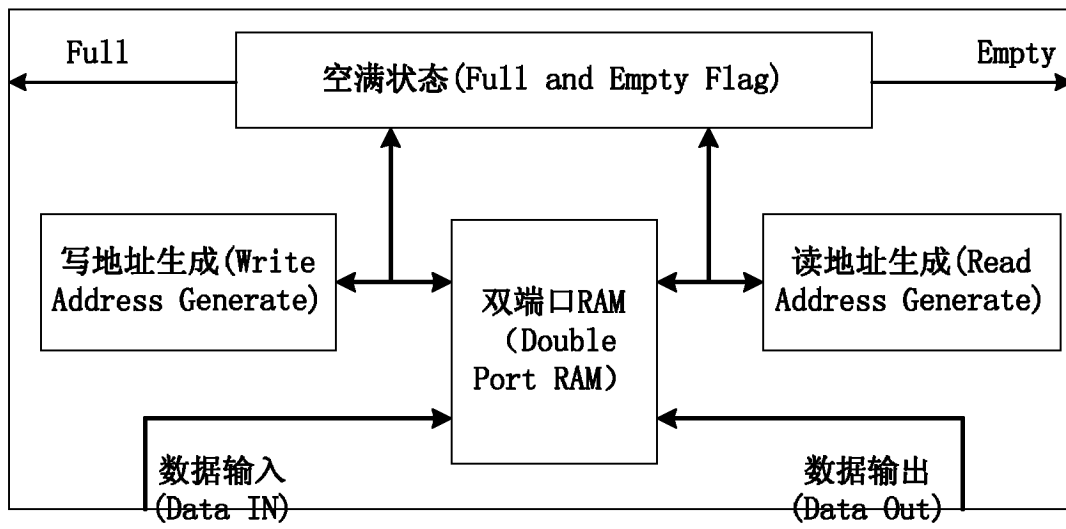


图 2

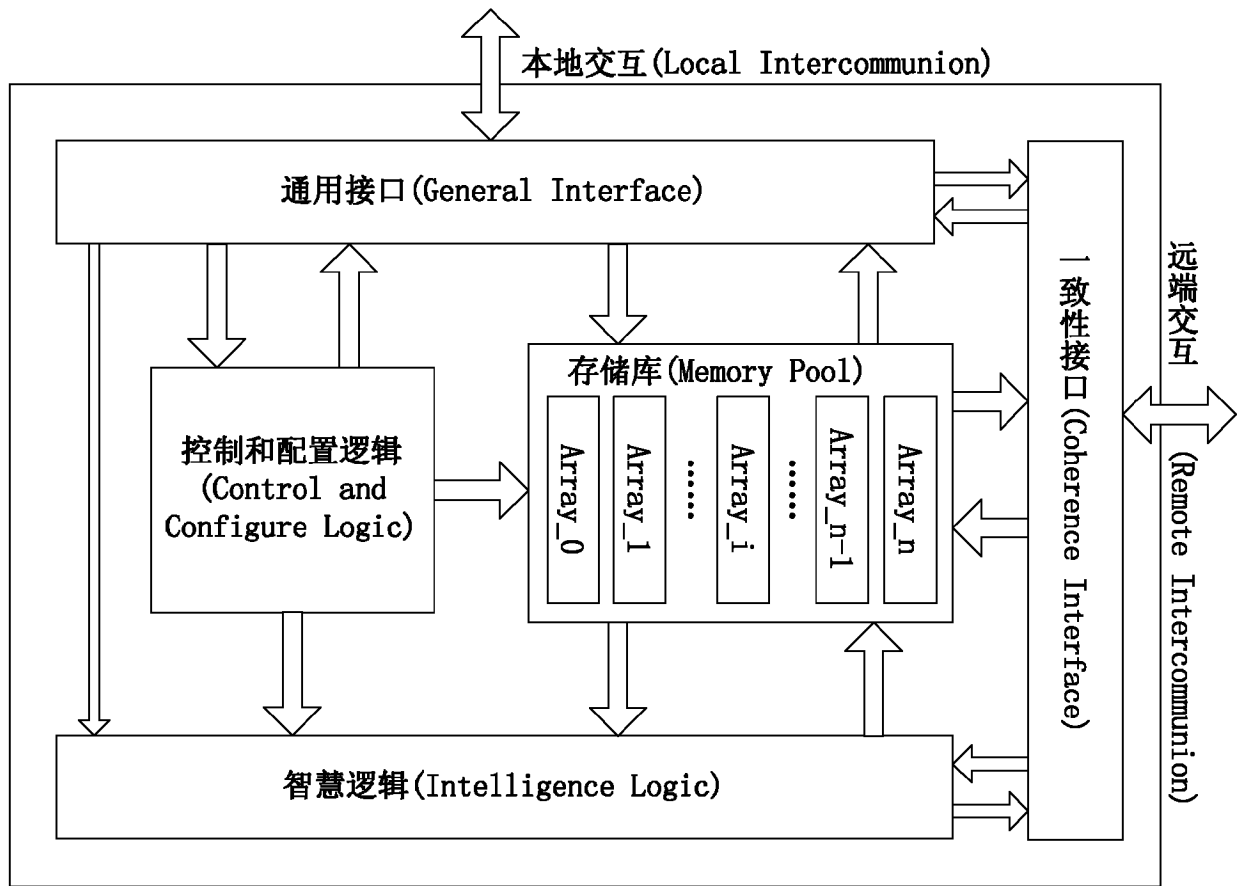


图 3

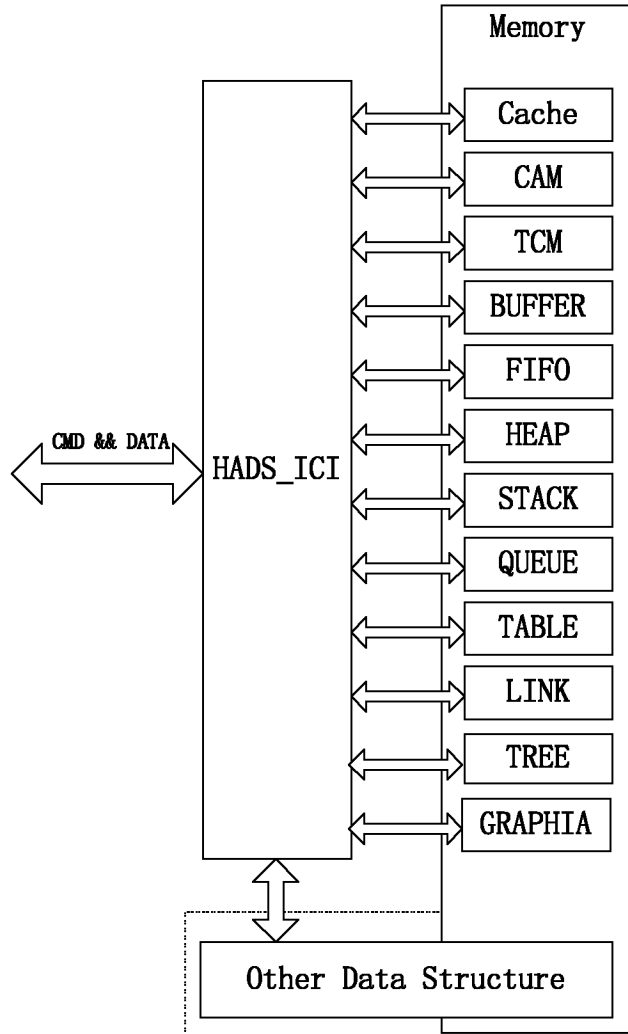


图 4

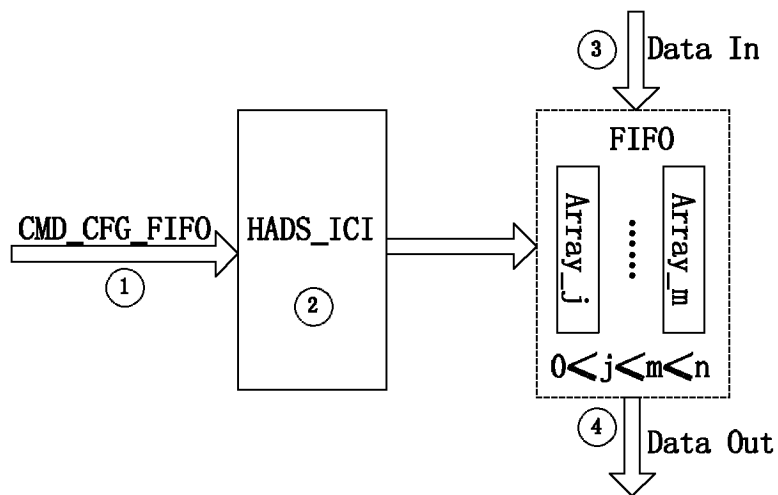


图 5

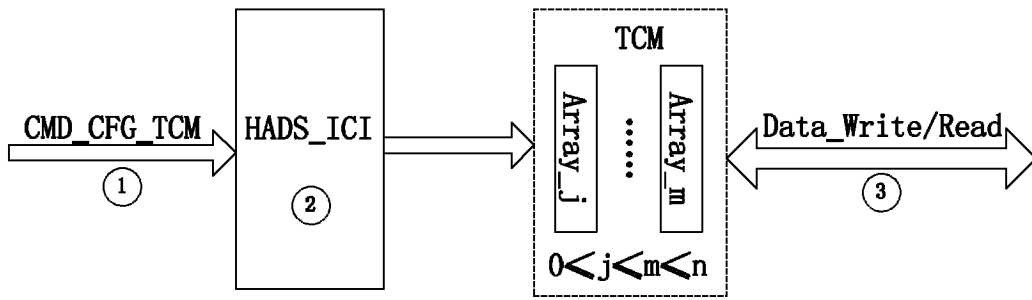


图 6

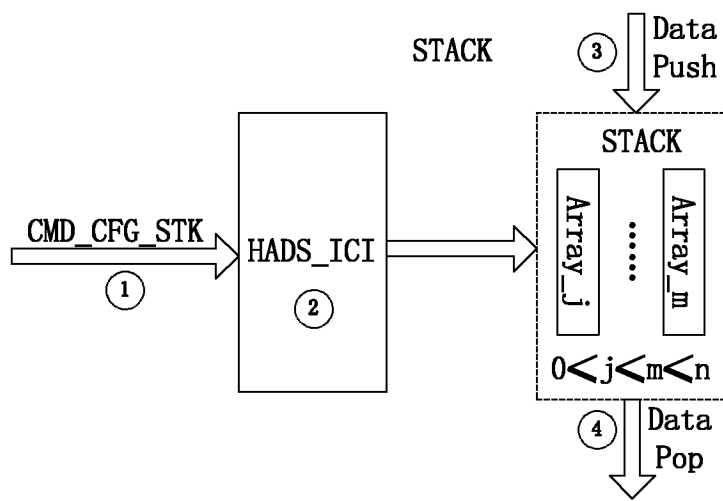


图 7

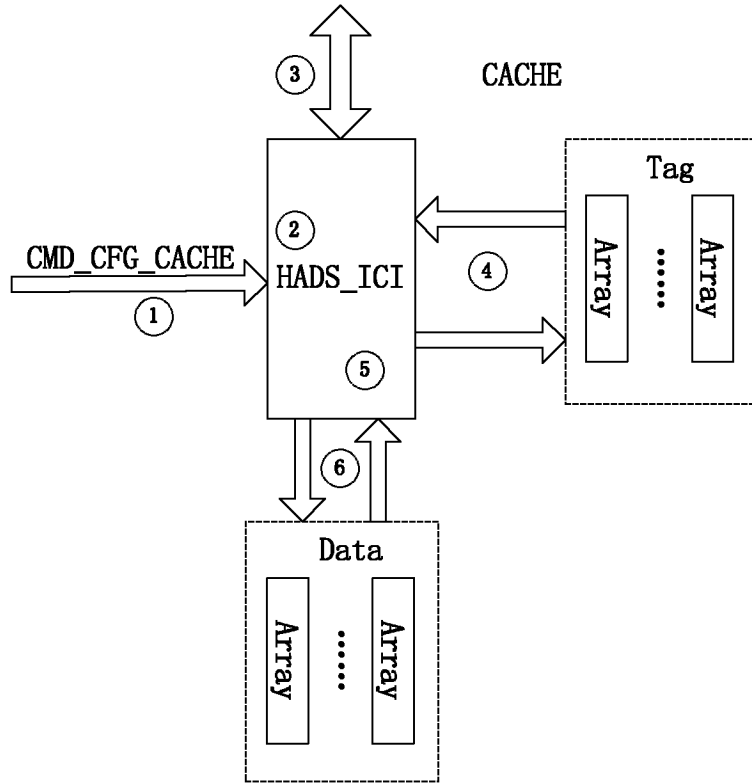


图 8

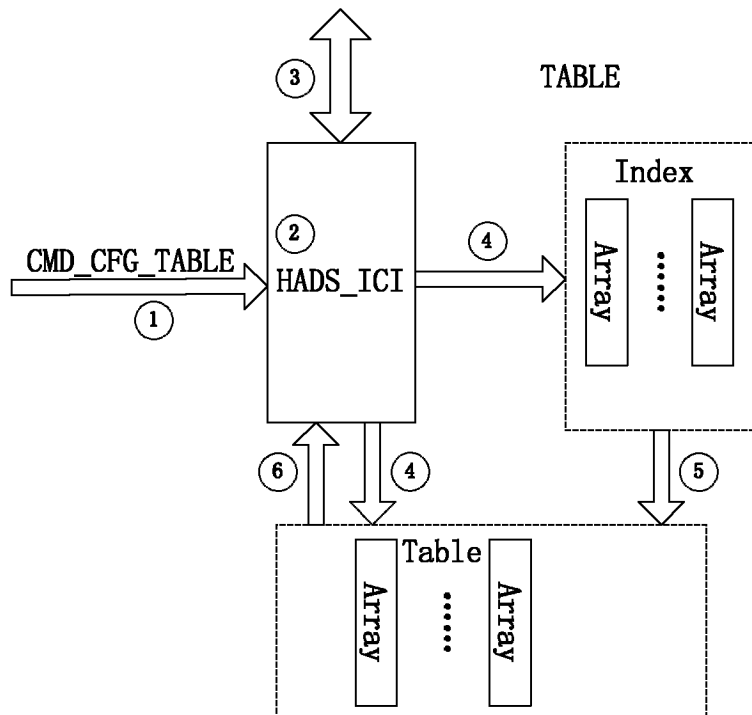


图 9

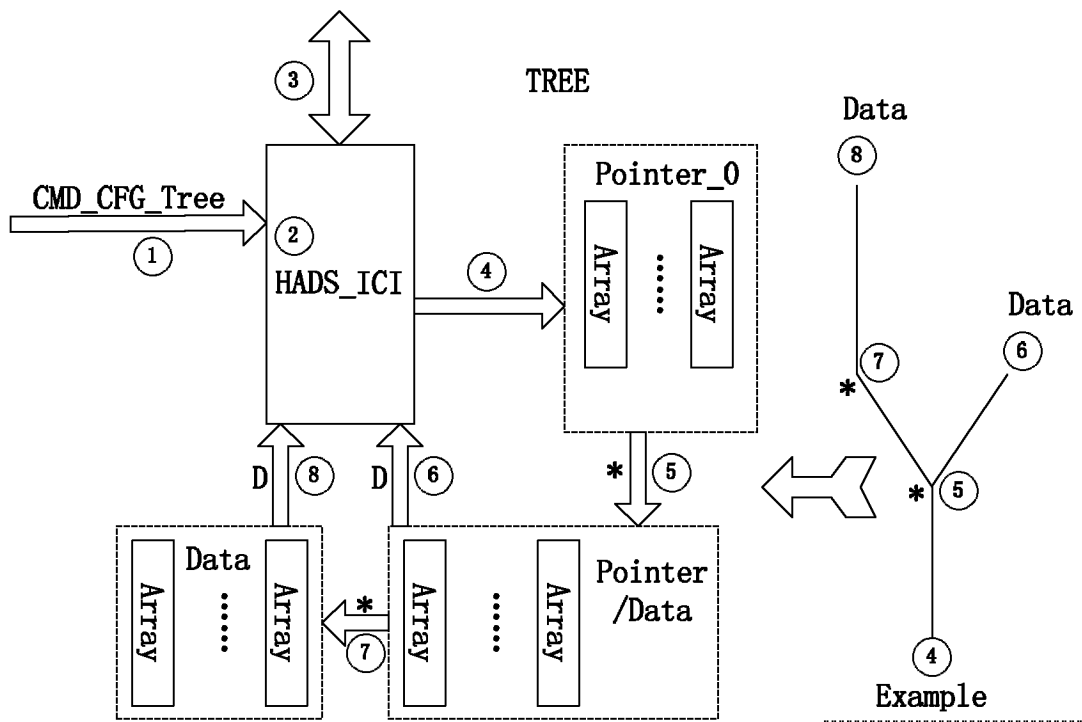


图 10

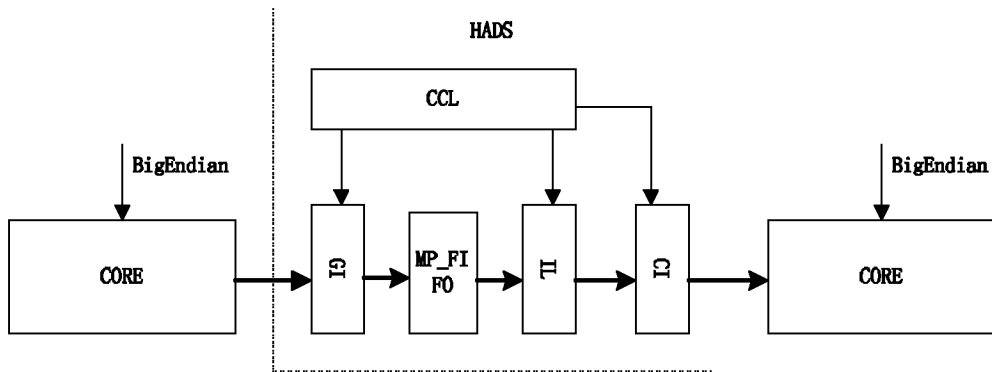


图 11

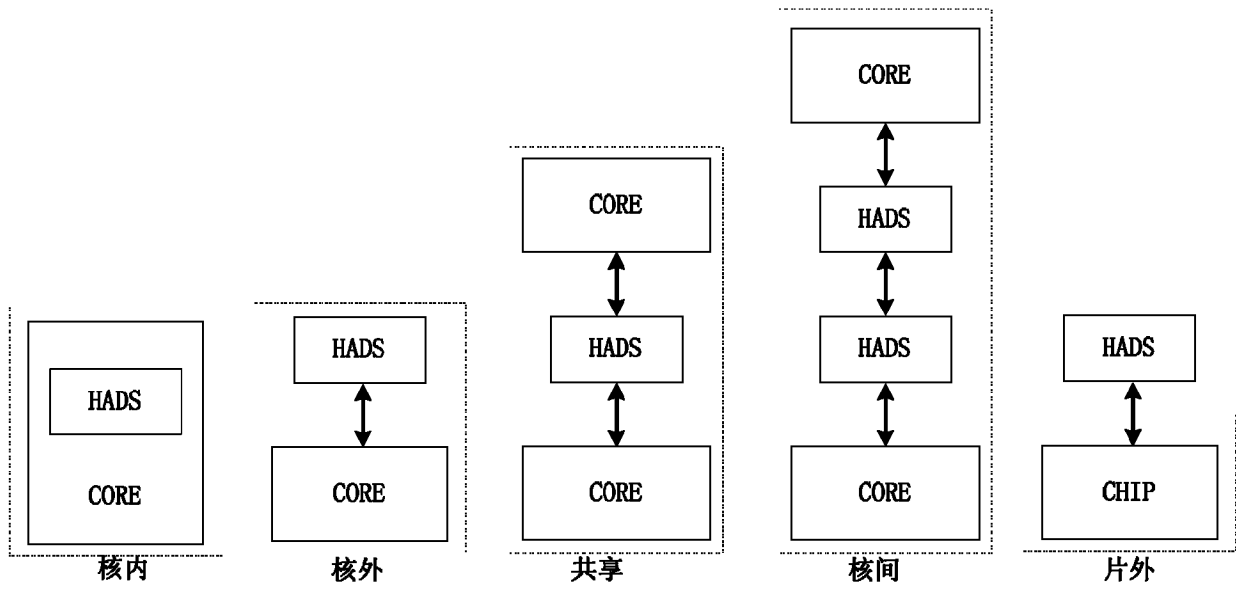


图 12

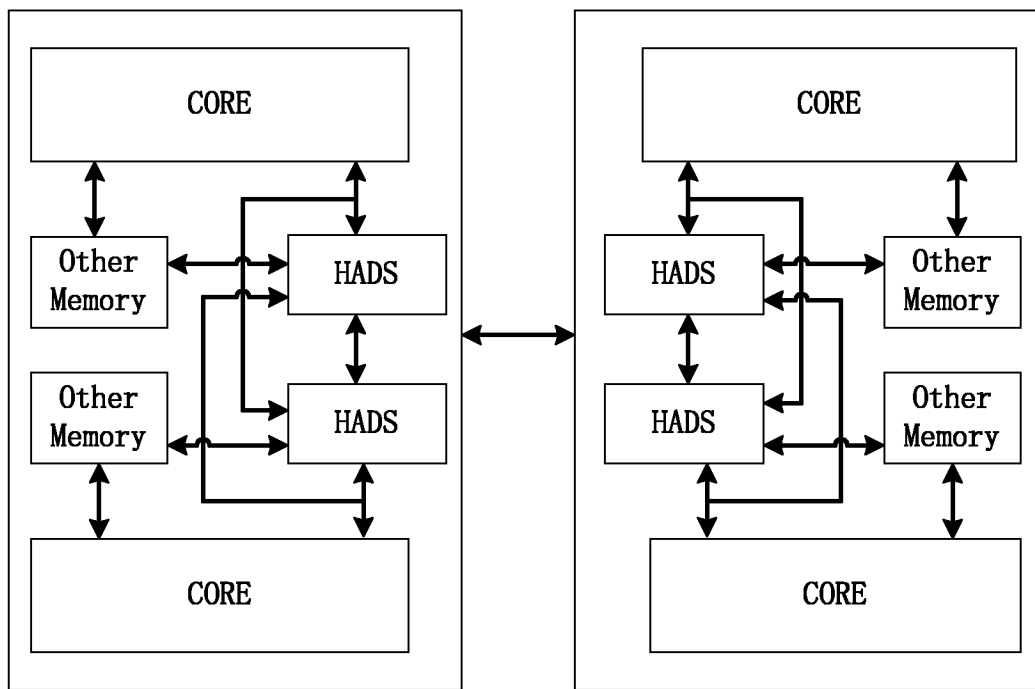


图 13