



(12)发明专利

(10)授权公告号 CN 104636268 B

(45)授权公告日 2019.07.26

(21)申请号 201310681802.9

(22)申请日 2013.12.11

(65)同一申请的已公布的文献号

申请公布号 CN 104636268 A

(43)申请公布日 2015.05.20

(66)本国优先权数据

201310556883.X 2013.11.08 CN

(73)专利权人 上海芯豪微电子有限公司

地址 200092 上海市杨浦区四平路1398号B
座1202

(72)发明人 林正浩

(51)Int.Cl.

G06F 12/0893(2016.01)

(56)对比文件

CN 101438237 A, 2009.05.20,

CN 101438237 A, 2009.05.20,

CN 102662868 A, 2012.09.12,

CN 101727406 A, 2010.06.09,

CN 102541510 A, 2012.07.04,

CN 1617112 A, 2005.05.18,

CN 102841777 A, 2012.12.26,

CN 102364431 A, 2012.02.29,

CN 101727406 A, 2010.06.09,

US 8271732 B2, 2012.09.18,

审查员 廖露露

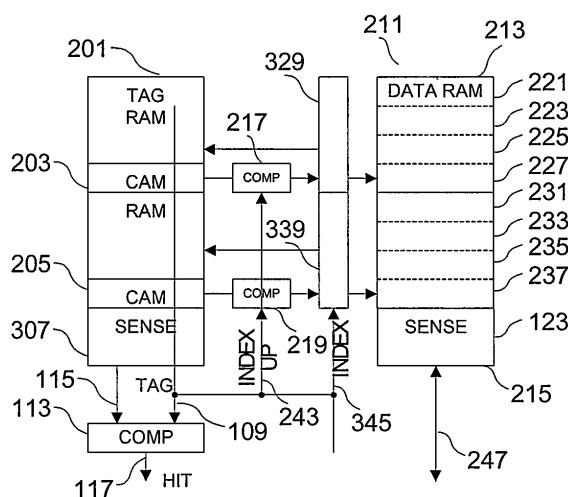
权利要求书2页 说明书12页 附图6页

(54)发明名称

一种可重构缓存产品与方法

(57)摘要

本发明提供了一种可重构缓存产品与方法，应用于处理器领域时能根据配置提供不同数目路组结构的缓存，并将连续地址的指令或数据存储在同一个路组中，便于处理器核获取指令或数据，以减少标签匹配次数。



1. 一种可重构缓存产品，其特征在于，缓存的指令或数据存储器中的多个存储块可以构成一个群 (group)，且对应同一个标签，形成按群可重构结构；当本次寻址地址与上次寻址地址对应同一个群时，可以省略本次寻址地址的标签比较，直接在该群中找到对应的指令或数据；

其路组相联结构可按需重构；其中每群存储块均可以被分配给任意一个组 (set) 构成其组中的一个路 (way)；且最大组数目与最大路数目的乘积大于存储块群的数目。

2. 根据权利要求1所述产品，其特征在于，所有存储块共享同一组位线。

3. 根据权利要求1所述产品，其特征在于，包括一个二维表；所述二维表的行对应组，且至少包含一行；所述二维表的列对应路，且至少包含一列；所述二维表的表项内容包括：标签、有效位和组号。

4. 根据权利要求3所述产品，其特征在于，根据存储器地址中的索引号从所述二维表中读出相应行中所有的有效标签值与存储器地址中的标签进行匹配；

根据匹配成功项中的组号在缓存中找到该存储器地址对应群；

根据存储器地址中的偏移地址在所述群中访问相应指令或数据。

5. 根据权利要求4所述产品，其特征在于，还包括至少一个独立标签模块；当一个独立标签模块被分配给一个组时，在所述独立标签模块中存储了所述组对应的索引号值和标签值。

6. 根据权利要求5所述产品，其特征在于，当需要对一个组分配新的路，且所述二维表中该组对应行中的全部表项均有效时，

若尚有未被占用的独立标签模块存在，再将该独立标签模块分配给该组使用；

若所有独立标签模块均被占用，则根据替换算法从部分或全部独立标签模块，以及所述二维表中该组对应的路中确定一个用于替换。

7. 根据权利要求6所述产品，其特征在于，当根据一个存储器地址在所述缓存中寻址时，根据所述存储器地址中的索引号从所述二维表的相应行中读出所有有效标签值，并从存储了所述索引号的所有独立标签模块中读出标签值与所述存储器地址中的标签进行匹配；

根据匹配成功项中的组号在缓存中找到该存储器地址对应群；

根据存储器地址中的偏移地址在所述群中访问相应指令或数据。

8. 一种可重构缓存方法，其特征在于，缓存的指令或数据存储器中的多个存储块可以构成一个群 (group)，且对应同一个标签，形成按群可重构结构；当本次寻址地址与上次寻址地址对应同一个群时，可以省略本次寻址地址的标签比较，直接在该群中找到对应的指令或数据；

其路组相联结构可按需重构；其中每群存储块均可以被分配给任意一个组 (set) 构成其组中的一个路 (way)；且最大组数目与最大路数目的乘积大于存储块的数目。

9. 根据权利要求8所述方法，其特征在于，所有存储块共享同一组位线。

10. 根据权利要求8所述方法，其特征在于，包括一个二维表；所述二维表的行对应组，且至少包含一行；所述二维表的列对应路，且至少包含一列；所述二维表的表项内容包括：标签、有效位和组号。

11. 根据权利要求10所述方法，其特征在于，根据存储器地址中的索引号从所述二维表

中读出相应行中所有的有效标签值与存储器地址中的标签进行匹配；

根据匹配成功项中的组号在缓存中找到该存储器地址对应群；

根据存储器地址中的偏移地址在所述群中访问相应指令或数据。

12. 根据权利要求11所述方法，其特征在于，还包括至少一个独立标签模块；当一个独立标签模块被分配给一个组时，在所述独立标签模块中存储了所述组对应的索引号值和标签值。

13. 根据权利要求12所述方法，其特征在于，当需要对一个组分配新的路，且所述二维表中该组对应行中的全部表项均有效时，

若尚有未被占用的独立标签模块存在，再将该独立标签模块分配给该组使用；

若所有独立标签模块均被占用，则根据替换算法从部分或全部独立标签模块，以及所述二维表中该组对应的路中确定一个用于替换。

14. 根据权利要求13所述方法，其特征在于，当根据一个存储器地址在所述缓存中寻址时，根据所述存储器地址中的索引号从所述二维表的相应行中读出所有有效标签值，并从存储了所述索引号的所有独立标签模块中读出标签值与所述存储器地址中的标签进行匹配；

根据匹配成功项中的组号在缓存中找到该存储器地址对应群；

根据存储器地址中的偏移地址在所述群中访问相应指令或数据。

一种可重构缓存产品与方法

技术领域

[0001] 本发明涉及计算机,通讯及集成电路领域。

背景技术

[0002] 为了提高性能,在现代处理器系统中通常采用缓存暂存主存储器中的指令或数据。存储在缓存中的指令或数据的地址一般分为三个部分:标签(TAG)、索引号(index)和块内偏移量(offset)。缓存一般由标签存储器和数据存储器构成,两者一一对应。其中数据存储器的每一个存储块存储了一个主存块(即指令块或数据块),标签存储器则存储了对应的存储块地址的标签。按组织形式的不同,缓存通常有三种形式:直接映射(direct map)、全相联(fully associative)和组相联(way-set associative)。

[0003] 在直接映射的缓存结构里,主存中的每个主存块在高速缓存中只能出现在唯一位置上。在直接映射的情况下,由于每个主存块在高速缓存中仅存在一个位置,因而地址的比较次数仅为一次。直接映射把主存储器分成若干页,主存储器的每一页与高速缓存的容量大小相同,且对应一个标签。根据主存块的地址中的索引号读出对应的标签,并与主存块地址中的标签部分比较,即可确定该索引号对应的存储块中存储的是否就是所述主存块地址对应的主存块。直接映射的缓存结构简单,实现容易,但每个索引号只对应一个存储块,因此直接映射的缓存中无法同时存储两个具有相同索引号的主存块。

[0004] 在全相联的缓存结构里,一个主存块可以被放到高速缓存中的任何一个地方,即存储块之间,以及存储顺序或保存的存储器地址之间没有直接的关系。高速缓存中可以保存互不相关的数据块,且必须对每个主存块及其自身的地址加以存储。当请求数据时,缓存控制器必须把主存块地址同所有存储在标签存储器中的块地址加以比较,进行确认。

[0005] 组相联是介于全相联和直接映射之间的一种结构。这种结构将高速缓存分成若干个路组(way-set),在路组内实现直接映射,而路组与路组间采用全相联方式。这样对于某一个给定的索引号,可以允许有几个块位置,因而可以增加命中率和系统效率。

[0006] 在上述三种组织结构中,全相联的缓存需要大量的比较器用于比较标签,电路结构复杂导致速度不高;且因所有比较器均同时进行比较,功耗非常大,在现代处理器的大容量缓存中无法实现。直接映射的缓存则因为每个索引号只对应一个存储块的局限,导致频繁替换,效率很低。只能有一个INDEX

[0007] 组相联的缓存则是现代处理器中最常用的缓存结构。通常路组数越多,缓存的性能越好,但需要同时读出、比较的标签也越多,导致功耗越高,且用于选择数据的选择器也越复杂,导致时延越大;敏感放大器和Y译码器所占的面积都很大,单靠分割位线(bit line)增加路组数代价很高。此外,有些程序由于其本身的特点,较低路组数的缓存就能实现不错的性能。因此,在组相联的缓存结构中,由于不知道会执行什么样的程序,因此很难在缓存性能与硬件成本之间作出平衡的选择。

[0008] 最后,在直接映射缓存结构及组相联结构的每个路组中,各个存储块的索引号是连续的,且每个索引号只能对应一个存储块,不够灵活。

[0009] 本发明提出的可重构缓存组织结构能直接解决上述或其他的一个或多个困难。

发明内容

[0010] 本发明提出了一种可重构缓存产品与方法,其特征在于,缓存的指令或数据存储器中的多个存储块可以构成一个组,且对应同一个标签,形成按组可重构结构;当本次寻址地址与上次寻址地址对应同一个组时,可以省略本次寻址地址的标签比较,直接在该组中找到对应的指令或数据。

[0011] 可选的,可以将所述缓存中的存储块配置为大小相等或不等的组。

[0012] 可选的,所述本次寻址地址和上次寻址地址是:两条地址连续的指令本身的地址;或两条被先后执行的数据访问指令分别对应的数据地址;或同一条数据访问指令先后被执行时分别对应的数据地址。

[0013] 可选的,当进行连续地址或相近地址寻址,且地址中的标签部分相同时,即可判定本次寻址地址与上次寻址地址对应同一个组。

[0014] 可选的,可以根据所述本次寻址地址与上次寻址地址中的索引号部分的差值,以及上次寻址地址对应的指令或数据在缓存中的位置,确定本次寻址地址对应的指令或数据在缓存中的位置。

[0015] 可选的,可以根据寻址地址中的标签、索引号高位部分的匹配结果,以及索引号低位部分的译码结果在缓存中找到相应指令或数据。

[0016] 可选的,可以通过选择线将寻址地址中的标签、索引号高位部分的匹配结果送往用于索引号低位部分译码的译码器;仅在匹配成功项对应的译码器中对索引号低位部分进行译码。

[0017] 可选的,可以先将选择线上的匹配结果编码后通过总线传输,经解码后再送到对应的译码器。

[0018] 可选的,还可以将所述缓存中的若干存储块配置为组相联结构,形成按组可重构结构及组相联结构共存的混合结构。

[0019] 可选的,所述缓存组织结构可以构成组相联缓存结构;其中每组(group)存储块均可以被分配给任意一个组(set)构成其的一个路(way);且最大组(set)数目与最大路(way)数目的乘积大于存储块组(group)的数目。

[0020] 可选的,在所述缓存组织结构中,所有存储块共享同一组位线。

[0021] 可选的,在所述缓存组织结构中,包括一个二维表;所述二维表的行对应组(set),且至少包含一行;所述二维表的列对应路(way),且至少包含一列;所述二维表的表项内容包括:标签、有效位和组号。

[0022] 可选的,在所述缓存组织结构中,根据存储器地址中的索引号从所述二维表中读出相应行中所有的有效标签值与存储器地址中的标签进行匹配;根据匹配成功项中的组号在缓存中找到该存储器地址对应组(group);根据存储器地址中的偏移地址在所述组(group)中访问相应指令或数据。

[0023] 可选的,在所述缓存组织结构中,还包括至少一个独立标签模块;当一个独立标签模块被分配给一个组(set)时,在所述独立标签模块中存储了所述组(set)对应的索引号值和标签值。

[0024] 可选的,在所述缓存组织结构中,当需要对一个组(set)分配新的路(way),且所述二维表中该组(set)对应行中的全部表项均有效时,若尚有未被占用的独立标签模块存在,在将该独立标签模块分配给该组(set)使用;若所有独立标签模块均被占用,则根据替换算法从部分或全部独立标签模块,以及所述二维表中该组(set)对应的路(way)中确定一个用于替换。

[0025] 可选的,在所述缓存组织结构中,当根据一个存储器地址在所述缓存中寻址时,根据所述存储器地址中的索引号从所述二维表的相应行中读出所有有效标签值,并从存储了所述索引号的所有独立标签模块中读出标签值与所述存储器地址中的标签进行匹配;根据匹配成功项中的组号在缓存中找到该存储器地址对应组(group);根据存储器地址中的偏移地址在所述组(group)中访问相应指令或数据。

[0026] 对于本领域专业人士,还可以在本发明的说明、权利要求和附图的启发下,理解、领会本发明所包含其他方面内容。

[0027] 有益效果

[0028] 本发明所述的可重构缓存组织结构可以提供一种可变大小的按组分配的缓存组织结构。每组中的存储块对应的主存地址是连续的,使得处理器核在获取连续地址的指令或数据时,所述缓存只需要经过简单的计算即可直接确定所述指令或数据在缓存中的位置,从而避免了标签匹配,降低了功耗。

[0029] 本发明所述的可重构缓存组织结构还可以按需配置为按组分配/组相联混合的缓存结构,将连续地址的指令或数据,以及不连续地址的指令或数据分别存储在按组分配的缓存部分及组相联结构的缓存部分中,使得缓存系统在兼容现有缓存结构的前提下,更好地支持连续地址的指令或数据读取。

[0030] 本发明所述的可重构缓存组织结构提供了一种介于组相联和全相联之间的缓存形式,可以根据程序运行时的客观需求,对不同索引号给予不同的路组数,用相当于组相联的硬件代价,实现了全相联的性能。

[0031] 对于本领域专业人士而言,本发明的其他优点和应用是显见的。

附图说明

- [0032] 图1是现有组相联缓存结构中标签比较的一个实施例;
- [0033] 图2是本发明所述可重构缓存中标签比较的一个实施例;
- [0034] 图3是本发明所述可重构缓存中标签比较的另一个实施例;
- [0035] 图4是本发明所述可重构缓存中一个组的实施例;
- [0036] 图5是本发明所述可重构缓存中标签比较的另一个实施例;
- [0037] 图6是本发明所述可重构缓存中标签比较的另一个实施例;
- [0038] 图7是本发明所述可重构缓存配置情况的一个实施例;
- [0039] 图8A是本发明所述可重构缓存的实施例;
- [0040] 图8B是本发明所述可重构缓存中地址格式和表项格式的一个实施例;
- [0041] 图8C是本发明所述可重构缓存的一个运行状态实施例;
- [0042] 图8D是本发明所述可重构缓存的另一个运行状态实施例;
- [0043] 图9是本发明所述可重构缓存的另一个实施例。

具体实施方式

[0044] 以下结合附图和具体实施例对本发明提出的高性能缓存系统和方法作进一步详细说明。根据下面说明和权利要求书,本发明的优点和特征将更清楚。需说明的是,附图均采用非常简化的形式且均使用非精准的比例,仅用以方便、明晰地辅助说明本发明实施例的目的。

[0045] 需要说明的是,为了清楚地说明本发明的内容,本发明特举多个实施例以进一步阐释本发明的不同实现方式,其中,该多个实施例是列举式并非穷举式。此外,为了说明的简洁,前实施例中已提及的内容往往在后实施例中予以省略,因此,后实施例中未提及的内容可相应参考前实施例。

[0046] 虽然该发明可以以多种形式的修改和替换来扩展,说明书中也列出了一些具体的实施图例并进行详细阐述。应当理解的是,发明者的出发点不是将该发明限于所阐述的特定实施例,正相反,发明者的出发点在于保护所有基于由本权利声明定义的精神或范围内进行的改进、等效转换和修改。同样的元器件号码可能被用于所有附图以代表相同的或类似的部分。

[0047] 本发明提供了一种可重构缓存产品与方法,应用于处理器领域时能根据配置提供不同数目路组结构的缓存,并将连续地址的指令或数据存储在同一个路组中,便于处理器核获取指令或数据,以减少标签匹配次数。在本说明书中,以数据缓存为例进行说明,但本发明所述结构和方法也适用于指令缓存。

[0048] 请参考图1,其为现有组相联缓存结构中标签比较的一个实施例。图中显示了一个路组的标签比较过程。其中,标签存储器101中的每行(1ine)对应一个索引号,其中存储了主存块地址中的标签部分。当需要从缓存中读取数据时,总线105上的数据寻址地址中的索引号107被送往译码器111,而标签109则被送往比较器113。

[0049] 译码器111对接收到的索引号107进行译码,得到相应的字线(word line)。所述字线被用于选中标签存储器101中与索引号107对应的行,并输出其中存储的标签经敏感放大器103放大后送往比较器113。比较器113比较敏感放大器103送来的标签115和标签109并通过总线117输出比较结果。若比较结果相等,说明在该路组中,标签115所在行对应的存储块(memory block)中存储的就是所述数据所在的数据块。若比较结果不相等,则表示在该路组中不存在所述数据所在的数据块。通过对缓存中所有路组进行如上操作,即可确定所述数据所在的数据块是否已经存储在缓存中,以及当已经存储在缓存中时的位置。这样,当比较结果为相等时,根据索引号107译码得到的相应字线从数据存储器121中读出的数据就是所述数据寻址地址对应的数据,并经敏感放大器123放大后经总线119送往处理器核。

[0050] 本发明所述的可重构缓存用新的结构和方法替代了图1实施例中的标签比较过程。请参考图2,其为本发明所述可重构缓存中标签比较的一个实施例。在本实施例中,每个路组中的标签存储器和数据存储器均被分为相应的若干组(group),每组对应连续索引号的相同数目行,且对应同一个标签。即,每个组中存储了对应同一个标签的连续地址的若干个数据块。

[0051] 以一个路组为例,如图2所示,标签存储器201被分为两个组,每组含一行内容寻址存储器(CAM),即存储一个标签(如标签203和标签205)。相应地,数据存储器211也被分为两个组,每个组含四个存储块,且这四个存储块中的数据块地址连续,并对应同一个标签。具

体地,组213中包含存储块221、223、225和227,这四个存储块中的数据块地址连续,且均对应标签203;组215中包含存储块231、233、235和237,这四个存储块中的数据块地址连续,且均对应标签205。需要说明的是,在本发明中,并不要求组213中四个数据块和组215中四个数据块的起始地址一致。在本实施例中,每组标签及相应组存储块还对应一个寄存比较器和一个译码器。如标签203对应寄存比较器217和译码器229,标签205对应寄存比较器219和译码器239。所述寄存比较器中包含一个寄存器和一个比较器。其中,所述寄存器存储了该组存储块起始地址中的索引号的高位部分。

[0052] 所述数据寻址地址中的标签部分被送到标签存储器201中的所有内容寻址存储器进行匹配,所有匹配成功的内容寻址存储器均向对应的寄存比较器中的寄存器输出一个使能信号。所述比较器则在该使能信号有效的情况下将外部送来的数据寻址地址中索引号的高位部分经总线243与所述寄存器中存储的索引号高位部分值进行比较,从而对相应组数据块地址进行部分(即所述索引号高位部分)匹配。所述译码器则在寄存比较器输出匹配成功的情况下,对总线245上的数据寻址地址中索引号的低位部分进行译码,并根据译码结果从相应组数据块中选出一个输出。这样,通过寄存比较器和译码器的匹配、译码及寻址,即可从数据存储器211中读出索引号与数据寻址地址中索引号相同的数据块。若所有内容寻址存储器均匹配不成功,或所有参与比较的比较器均匹配不成功时,则说明所述数据寻址地址对应的数据尚未存储在缓存的该路组中。这样,按同样方法并行对所有路组进行如上操作,即可在缓存中找到所需数据经敏感放大器123放大后由总线247输出,或得到缓存缺失的结果。

[0053] 由于内容寻址存储器的比较线(Match Line)必须先充电,才能进行标签匹配,而充电、匹配过程耗电较多,在本实施例中同时在所有内容寻址存储行中匹配标签功耗较大。因此,还可以改进地址匹配的顺序,以进一步降低功耗。具体地,先将所述数据寻址地址中的索引号高位部分通过总线243送到所有寄存比较器中与存储的索引号高位部分值进行比较,并根据比较结果,只对匹配成功项对应的内容寻址存储行的比较线进行充电、与经总线241送来的标签匹配,并由匹配成功的內容寻址存储行向译码器输出使能信号。后续操作与前述相同。这样,减少了内容寻址存储行的匹配数目,降低了功耗。

[0054] 在本实施例中,只需要对寄存比较器中的寄存器存入相应的索引号高位值,即可对缓存进行重构。例如,可以在两个相邻的寄存比较器中存入连续的索引号高位值,使得这两个寄存比较器对应的索引号也连续。这样,所述相邻的两个组就被合并为一个更大的组以容纳连续地址的数据块。

[0055] 请参考图3,其为本发明所述可重构缓存中标签比较的另一个实施例。同样以一个路组为例,其结构与图2实施例中的类似,不同之处在于本实施例中的标签存储器301的每组不但有一行内容寻址存储器,还包含了随机访问存储器(RAM)。其中,标签存储器301每一组中的内容寻址存储器与数据存储器211中相应组内的第一个存储块对应,而其他的各行随机访问存储器分别与相应组内的其他存储块对应,使得数据存储器211中的每个存储块都有对应的标签。此外,在本实施例中,译码器(如译码器329和339)可以根据需要,对完整的索引号进行译码,或如图2实施例中的译码器229和239那样,只对索引号低位进行译码。

[0056] 在本实施例中,所述可重构缓存就可以按需被配置为传统形式的组相联缓存,或本发明所述的按组可重构缓存。当配置为组相联缓存时,每个路组中的存储块对应连续的

索引号,但不必对应相同的标签。此时,标签存储器中的内容寻址存储器被视为随机访问存储器,和其他随机访问存储器一起分别存储了对应的存储块中数据块的标签。同时,每个寄存比较器(如寄存比较器217、219)的输出被配置为固定输出‘1’。这样,数据寻址地址中的索引号经译码器的匹配、译码和寻址后,就可以在标签存储器301和数据存储器211中分别找到该索引号对应的标签和存储块。所述标签经敏感放大器307放大后在比较器113中与数据寻址地址中的标签比较及后续过程与图1实施例中相同,从而确定所述数据所在的数据块是否已经存储在缓存中,以及当已经存储在缓存中时读出对应数据块。

[0057] 当配置为按组可重构缓存时,每个组内的随机访问存储器被忽略,每个寄存比较器中的寄存器都存放了索引号高位部分,而译码器只对索引号的低位部分进行译码,这样本实施例所示结构就具备了和图2实施例的相同功能,即可按图2实施例所述方法进行操作。

[0058] 请参考图4,其为本发明所述可重构缓存中一个组的实施例。在本实施例中,假设索引号共有4位,其中高位部分和低位部分均为2位。

[0059] 图4中所显示的就是图3中一个组(如标签205对应的组)的具体结构。其中,标签存储器部分包含一个内容寻址存储行401(即存储了标签205)和三个随机访问存储行403、405和407。数据存储器部分则相应地存储了四个存储块411、413、415和417。译码器429和439分别对索引号的高位部分和低位部分进行译码,两者一同实现了图3实施例中的译码器239的功能。

[0060] 在本实施例中,当该组缓存被配置为组相联缓存时,开关421、423、425和427均呈如图所示形式,即将译码器409中各个与门的输出连通到相应的标签存储器行中。此时,如图3实施例所述,比较器219的输出固定为‘1’,其结果被送到本组中的所有与门作为一个输入。索引号的高位部分经总线451由译码器429译码后选中一个组(即向该组中的所有与门输出一个有效信号),而低位部分则经总线453由译码器439译码后选中该组中的一个与门,即给予该与门一个有效信号(‘1’),并给予其他所有与门一个无效信号(‘0’)。假设在此情况下,译码器429选中图4所示的组,且译码器439选中与门433,则在与门433输出信号的控制下,随机访问存储行403中的标签被读出并经敏感放大器307放大后在比较器113中与数据寻址地址中的标签进行比较,并从存储块413中读出对应的数据块送往选择/放大模块419。若本路组比较相等,则说明存储块413中读出的数据块就是该数据寻址地址对应的数据块,根据总线441上的数据寻址地址中的块内偏移量即可选出对应的数据放大并从总线445输出。若本路组比较不相等,但其他某个路组输出的标签比较相等,则那个路组输出的数据块就是该数据寻址地址对应的数据块,根据数据寻址地址中的块内偏移量即可选出对应的数据。若所有路组比较都不相等,则说明该数据寻址地址对应的数据块尚未存储在缓存中。

[0061] 当该组缓存被配置为按组可重构缓存时,开关421、423、425和427均接地,即将各个与门的输出与相应的标签存储器行断开,且对所有标签存储器行均不做选取;并强制译码器429的所有输出固定为‘1’。此时,如之前实施例所述,数据寻址地址中的索引号高位部分经总线243被送到寄存比较器219进行匹配,并对匹配成功项对应的内容寻址存储行充电,使该内容寻址存储行中存储的标签与由位线455送来的数据寻址地址的标签进行匹配。若标签匹配不成功,则本组中没有包含所述数据寻址地址对应的数据。若标签匹配成功,则

输出有效信号443到本组中的各个与门。此时,译码器429的输出固定为‘1’,且索引号的低位部分经经总线453由译码器439译码后选中一个与门。假设在此情况下,比较器219及译码器429均选中图4所示的组,且译码器439选中与门435,则在与门435输出信号的控制下,读出存储块415中的数据块送往选择/放大模块419。选择/放大模块419则直接根据数据寻址地址中的块内偏移量441,直接从该数据块中选出对应的数据并由总线445输出。在此过程中,标签存储器内各个随机访问存储行均没有参与操作。虽然本实施例以数据读取为例进行说明,但对于数据存储而言,也可以用类似方法实现。不同之处仅在于进行数据读取时,从数据缓存211中匹配的存储块中经位线457读出数据块经放大选择后输出相应数据,而进行数据存储时,直接将需要存储的数据(或数据块)经位线457写入数据缓存211中匹配的存储块中。

[0062] 此外,还可以将所述缓存中的译码器(如译码器329和339)改为只指向数据存储器211中的存储块,并增加一套译码器用于指向标签存储器201中的内容寻址存储行。这样,如果标签存储器201和数据存储器211物理位置相隔较远时,依然可以实现本发明所述缓存的功能。

[0063] 请参考图5,其为本发明所述可重构缓存中标签比较的另一个实施例。在本实施例中,增加了译码器529和539。其中,译码器529的结构和功能与译码器329相同,两者均接收寄存比较器217送出的比较结果;译码器539的结构和功能与译码器339相同,两者均接收寄存比较器219送出的比较结果。这样就可以不必如图4实施例那样将译码器329和339输出的字线同时送往数据存储器211,并经开关(如开关421)送往标签存储器201。以译码器329及对应的译码器529为例,当所述缓存被配置为组相联缓存时,译码器329和译码器529同时对数据寻址地址中的索引号译码,并分别根据译码输出的字线从数据存储器201读出数据块及从标签存储器201中读出标签,以进行后续操作。当所述缓存被配置为按组可重构缓存时,译码器529不工作,按前述方法仅由相应的寄存比较器及内容寻址存储行进行匹配,以确定数据在哪个组中,并将匹配结果通过总线送往相应译码器(即,寄存比较器217及内容寻址存储行203的匹配结果经选择线517送往译码器329);并根据译码器329对总线345上的数据寻址指令中索引号的低位部分,译码得到的字线在该组中找到对应的数据块,以进行后续操作。其他的译码器(如译码器339和539)也按相同方法运行,例如寄存比较器219及内容寻址存储行205的匹配结果经选择线527送往译码器339等。当因缓存被分为很多组,导致所述传递到译码器的匹配结果很多时,可以先将所述匹配结果进行编码后通过总线传输到数据存储器一侧,经解码后再分别送往对应的译码器,从而减少选择线(如选择线517、527)的数目。例如,16条选择线可以由4条总线代替。

[0064] 需要说明的是,在图2、图3和图4实施例中,采用寄存比较器的结构实现了对索引号高位部分的匹配。当然也可以用内容寻址存储器代替所述寄存比较器实现同样功能。对于图2实施例,只需要增加标签存储器201中的每个内容寻址存储器的位宽,使得标签和索引号高位部分都能被存储在所述内容寻址存储器并参与匹配,即可实现所述功能。对于图3和图4实施例,则在增加标签存储器301中的每个内容寻址存储器的位宽的同时,保持随机访问存储器不变即可。请参考图6,其为本发明所述可重构缓存中标签比较的另一个实施例。仍然以一个路组为例,本实施例与图3实施例结构相似,不同之处在于增加了标签存储器301中的每个内容寻址存储器的位宽,以代替寄存比较器。内容寻址存储器增加的部分中

存储了就是相应的索引号高位部分。

[0065] 在本实施例中,当缓存被配置为组相联缓存时,数据寻址地址中的索引号的高位部分经位线243被送到标签存储器481中所有内容寻址存储器新增加的部分进行匹配,而匹配成功项对应的译码器则对索引号的低位部分进行译码,从而在标签存储器481和数据存储器211中分别找到该索引号对应的标签和存储块。后续操作过程与图3实施例中相同,在此不再赘述。当缓存被配置为按组可重构缓存时,每个组内的随机访问存储器被忽略,数据寻址地址中的标签和索引号高位部分被送到标签存储器481各个内容寻址存储器匹配,并由匹配成功项对应的译码器对索引号低位部分译码。后续操作过程与图3实施例中相同,在此不再赘述。这样,采用本发明所述缓存结构,可以在不增加敏感放大器和Y译码器的情况下,很方便地将一个路组中的存储块分割为多个组,每个组起到了一个路组的作用。

[0066] 请参考图7,其为本发明所述可重构缓存配置情况的一个实施例。为了便于说明,在图7中只显示了标签存储器和译码器,省略了其他部分。本实施例的缓存具有四个路组(如路组501、503、505和507),每路则被分为四个组(如路组501中的组511、513、515和517),每组中则可以包含若干个用于存放数据块的存储块。因此,在所有16个组中就可以最多存放对应16个标签的连续地址的数据块。

[0067] 根据本发明技术方案,可以用路组号及路组内号码一同构成一个对标签存储器(及数据存储器)中的每个组进行唯一识别的组号码。如图7中标签存储器每组上的数字所示,前两位为路组号,后两位为路组内的组号。即,路组501的路组号为‘00’,路组503的路组号为‘01’,路组505的路组号为‘10’,路组507的路组号为‘11’。而每个路组中的各个组从上往下的路组内号码依次为‘00’、‘01’、‘10’和‘11’。因此,组511对应的组号码就是‘0000’,组513对应的组号码就是‘0001’,组515对应的组号码就是‘0010’,组517对应的组号码就是‘0011’,以此类推。这样,当一个数据寻址地址在某组中匹配成功时,就可以用相应的组号码及组内偏移量(用以指明组内的哪个数据块以及该数据块中的哪个数据)代替该数据寻址地址,使得下一次重复访问该数据时不必再次进行标签匹配。

[0068] 进一步地,当程序中的数据读取指令(或数据存储指令)依次访问相邻或相近位置的数据时,可以根据前后两个数据寻址地址的差值,及前一个数据寻址地址对应的组号码,直接推算出本次数据寻址地址对应的组号码,以避免标签匹配。

[0069] 例如,假设前一次数据读取指令的数据寻址地址对应的组号码是‘1110’,即所述数据寻址地址对应的数据位于组545对应的存储块中。将本次数据读取指令的数据寻址地址中的标签和索引号分别减去所述前一次数据寻址地址中的标签和索引号。若标签相减的结果为‘0’,即表示本次数据寻址地址与上次数据寻址地址的标签相同,因此位于同一个组545内,即本次数据寻址地址对应的组号码也是‘1110’。根据索引号相减的结果则可以确定本次数据寻址地址对应存储块与上次数据寻址地址对应的存储块之间的位置关系。具体地,若索引号相减的结果为‘0’,则本次数据寻址地址对应存储块就是上次数据寻址地址对应的存储块,并可根据数据寻址地址中的块内偏移量从该存储块中选出对应的数据;若索引号相减的结果为正,则本次数据寻址地址对应存储块位于上次数据寻址地址对应的存储块之后;若索引号相减的结果为负,则本次数据寻址地址对应存储块位于上次数据寻址地址对应的存储块之前。对于后两种情况,索引号相减得到的差值的绝对值就是所述两个存储块的间隔距离。例如,索引号相减得到的差值为‘2’,表示本次数据寻址地址对应的存储

块位于上次数据寻址地址对应的存储块之后的第二个存储块。又如,索引号相减得到的差值为‘-1’，表示本次数据寻址地址对应的存储块位于上次数据寻址地址对应的存储块之前的第一个存储块。这样,不需要经过任何标签比较,就可以确定本次数据寻址地址对应的数据块在缓存中的位置。

[0070] 此外,当连续的数据块超过单个组的容量时,还可以将这些数据块存储在组号码连续的两个组中。例如,当组513和515对应同样的标签,且组515对应的索引号高位部分等于组513对应的索引号高位部分加‘1’时,不但组513和515中各自存储的数据块的地址是连续的,组513中最后一个数据块和组515中第一个数据块的地址也是连续的。这样,构成了组与组之间的连接关系,即组513是组515的前导(previous)组,而组515是组513的后续(next)组,从而能够根据索引号差值直接从一个组找到位于另一个组中的数据位置。

[0071] 当然,若标签相减的结果不为‘0’,则可以使用之前实施例所述方法,在标签存储器中对本次数据寻址地址进行匹配,以找到对应的组。

[0072] 此外,在本发明中,还可以将各个组配置成不同大小,例如可以将图7实施例中的路组501配置成四个组(即组511、513、515和517),并将路组503配置成一个组,以及将路组505和507配置成传统形式的组相联结构。在这种情况下,路组501中最多包含四个不同的标签,而路组503中只包含一种标签。路组505和507则如现有二路组缓存那样,各自可包含的最多标签数等于对应的存储块的数量(及路组本身的行数),相邻的存储块可以对应不同的标签。采用如此配置的缓存,可以根据程序的特点,将数据寻址地址连续(即标签相同)的大量数据存储在路组503中,并将多组数据寻址地址连续的少量数据存储在路组501的各个组中。对于数据寻址地址不连续的数据,则被存储在路组505或507中。这样,所述缓存即具备了缓存中数据存放的灵活性和便于替换的特点,又能在进行连续地址的数据访问时省去大量的标签比较操作。

[0073] 以上的实施例均针对数据缓存进行说明。由于指令缓存的结构与数据缓存相似,因此完全可以用同样的结构实现,在此不再赘述。

[0074] 根据本发明技术方案,还可以对缓存中的标签部分进行改进,实现路组数可变的可重构缓存。请参考图8A,其为本发明所述可重构缓存的实施例。在本实施例中,可重构缓存由标签部分801和数据存储部分803组成,其中数据存储部分803中的所有存储块共享同一组位线(bit line),并按之前实施例中的数据存储器211的分组方法,被分为6个组(group):A、B、C、D、E和F,因此组号为3位,例如:组A对应组号‘000’、组B对应组号‘001’、.....、组F对应组号‘101’。每个组(group)中包含多个存储块,而每个存储块中包含多条指令或多个数据。标签部分801中则以一个二维表825的形式存储了各个组(group)对应的标签、组号等信息。如图8B所示,表项内容807可以包含标签811、有效位817和组号819。在本发明中,有效位817为‘1’表示该表项是有效表项;有效位817为‘0’表示该表项是无效表项。此外,二维表825的列数对应所述缓存所支持的最大路数,行数则表示同一路(way)中最大支持的组数,即同一路(way)中最大支持的索引号值。在此,二维表825有4行4列,其中4列表示所述缓存最多支持4个路(way);4行则表示缓存可以分为4个组(set),索引号的范围为‘0’到‘3’,即索引号的位数为2位。存储器地址中的索引号与缓存中的组(set)一一对应。

[0075] 此外,比较模块821由多个比较器构成,比较器的数目与二维表825的列数相等。比

较模块821对标签部分801中根据索引号813读出的所有有效标签与存储器地址805中的标签811进行比较,比较结果则被送到选择模块823作为控制信号。选择模块823由多个传输门组成,传输门的数目与二维表825的列数相等。每个传输门的输入为对应的二维表列输出的表项内容中的组号819。这样,根据所述比较结果,即可选出匹配成功表项对应的组号819。

[0076] 如图8B所示,存储器地址805被分为三个部分:标签811、索引号813和偏移地址815。其中,索引号813对应二维表825的行数,即位数为2位;而偏移地址815对应指令或数据在组(group)内的位置,其位数固定。例如,假设每个组(group)包含8个存储块,每个存储块包含16个字节指令或数据,对于32位长的存储器地址,偏移地址815共7位(即每组8个存储块共128字节指令或数据);索引号813为2位,其余23位为标签。

[0077] 这样,根据所述索引号813读出所述二维表中对应行中存储的所有有效表项,并将这些表项中存储的标签与存储器地址805中的标签811同时比较。若均不匹配,则表示缓存缺失。若有匹配,则匹配项中存储的组号就是所述存储器地址805对应的组(group)。即,将存储器地址805中的标签811和索引号813转换为组号819,从而由组号819和偏移地址815构成缓存寻址地址809。根据寻址地址809中的组号819即可在缓存中找到相应的组(group),并根据寻址地址809中的偏移地址815从该组(group)中访问相应指令或数据。

[0078] 需要注意的是,现有的组相联缓存中,每个索引号对应的标签存储位置的数目相同,即等于路(way)数。例如,在一个4路(way)缓存中,索引号M和索引号N都对应4个标签。但在本实施例的缓存结构中,每个索引号对应的标签存储位置的数目可以不同。例如,在本实施例所述的最多支持4路(way)的缓存中,索引号M可以只对应2个标签,即2路(2way);但索引号N对应4个标签,即4路(4way)。

[0079] 以下结合图8A,以具体实施例进行说明。请参考图8C,其为本发明所述可重构缓存的一个运行状态实施例。在本实施例中只显示了标签部分801中存储的标签值。在本实施例中,假设在某一时刻,所述缓存已经为索引号‘00’分配了四个组(group):B、C、D、E,并为索引号‘10’分配了组A;组F尚未被分配。当根据存储器地址805寻找对应指令或数据时,根据其中索引号813读出二维表825中相应行的所有表项内容并将其中的标签和有效位送往比较器模块821。其中,接收到的有效位为‘0’的比较器的比较输出均为‘0’(即不匹配);接收到的有效位为‘1’的比较器对接收到的标签与存储器地址中的标签811进行比较,并输出比较结果。

[0080] 具体地,若存储器地址805中的索引号值是‘01’或‘11’,则二维表825中的对应行没有有效表项,因此均不匹配,比较模块821输出缓存缺失信号到总线829上。以该索引号值是‘11’为例,缓存分配一个可用组F给索引号‘11’。即,将该存储器地址805中的标签及组号F存储到二维表825第2行中的一个无效表项中,并将该表项设为有效。此时,二维表825中的状态如图8C所示。

[0081] 之后,若存储器地址805中的索引号值是‘00’,则对应的第0行中的所有表项内容均被读出。由于这些表项均有效,因此这些标签被送往比较器模块821分别与存储器地址中的标签811进行比较,所有比较器的比较结果由逻辑门827进行‘或’操作后输出。同时读出内容中的组号(B、C、D或E)则被送往选择模块823作为传输门的输入。这样,若匹配成功,则比较模块821输出缓存命中信号到总线829上,且选择模块823输出匹配成功项对应的组号819输出。所述组号819与偏移地址815一同构成缓存寻址地址809,之后利用该寻址地址809

即可直接从数据存储部分803中访问相应指令或数据。

[0082] 若匹配均不成功，则比较模块821输出缓存缺失信号到总线829上。由于索引号‘00’对应的4个路(way)均已被占用，因此可以根据替换算法(如LRU算法)，从对应的4个组(group)：B、C、D、E中选择一个合适的组(group)分配给该存储器地址805使用。

[0083] 若存储器地址805中的索引号值是‘10’，则对应的第2行中的所有表项内容被读出，其中的有效位和标签被送往比较器模块821，组号则被送往选择模块823作为相应传输门的输入。由于只有一个表项有效，若该表项中的标签匹配成功，则比较模块821输出缓存命中信号到总线829上，且选择模块823输出该表项对应的组号A。所述组号与偏移地址815一同构成缓存寻址地址809，之后利用该寻址地址809即可直接从数据存储部分803中访问相应指令或数据。

[0084] 若该表项中的标签匹配不成功，则比较模块821输出缓存缺失信号到总线829上。由于索引号‘10’对应的路(way)中只有1个被占用，因此可以根据替换算法(如LRU算法)，从所有6个组(group)中选择一个合适的组(group)分配给该存储器地址805使用。此时，若选择得到的被替换组(group)不是组A(例如是组C)，则该组(group)原对应的索引号(例如索引号‘00’)相应减少1路(way)，而索引号‘10’相应增加1路(way)。这样，可以在程序运行过程中根据程序需求，自动地对不同索引号给予不同数目的路(way)，灵活分配缓存资源，从而提高缓存命中率。

[0085] 请参考图8D，其为本发明所述可重构缓存的另一个运行状态实施例。在本实施例中，假设在某一时刻，所述缓存已经为每个索引号均分配了一个组(group)，分别是D、A、B、C；组E、F尚未被分配。这样，当根据存储器地址805寻找对应指令或数据时，如果索引号813对应的标签匹配成功，则可按之前实施例所述方法由选择模块823输出对应的组号。若匹配不成功，则根据替换算法分配一个可用组(group)给该索引号813。例如，根据程序运行需求，先后分配了组E和F给索引号‘01’和‘10’使用，使得二维表825的状态如图8D所示。本实施例中的具体运行过程与图8C实施例中类似，在此不再赘述。

[0086] 传统的缓存其地址标签部分与数据存储部分对称(Symmetrical)且对应，其结构是固定的。如：一共有8个存储块组(group)，其组织方式可以是1路8组，2路4组，4路2组，以及8路1组(全相联)；但是其组织形式是固定的，在设计时选定后即不可更改。

[0087] 本实施例所示的缓存其地址标签部分与数据存储部分不对称，成可变的映射关系，因而其结构不是固定的。其地址标签部分的地址空间大于缓存存储块组能占据的空间。如：数据存储部分803在传统缓存中固定是一个1路6组(1way, 6set)的存储器。在本实施例中，该6个存储块组(group)可以按需要被映射到标签部分801中4路4组(4way, 4set)共16个地址位置中，构成不同组合的多种路组缓存器。在本实施例中，组(set)可以被转换成路(way)。另一方面，如果有两个数据存储部分803，在传统缓存内为2路(way)，那么分别在两个不同数据存储部分中的存储块组(group)可以被映射到同一路(way)中的不同组(set)。即，路(way)被转换为组(set)。这样，组(set)和路(way)可以相互交换，因此可以将这视为一种新型的路-组交换缓存(way-set exchange cache)。如果二维表825再增加2列，并在比较模块821和选择模块823中增加相应的比较器和选择器，则可以按需实现6路1组(6way, 1set)的全相联缓存。

[0088] 此外，还可以用更灵活的结构实现所述可重构缓存。请参考图9，其为本发明所述

可重构缓存的另一个实施例。本实施例中的数据存储部分与图8A实施例的相同,也包含6个组(group)。为了便于说明,在图9中只显示了标签部分。本实施例所述标签部分中的比较器921和选择器923分别与图8A实施例中比较模块821中的比较器及选择模块823中的选择器相同。图8A实施例中的二维表825包含译码器901和4个标签列,每列有4个表项,而本实施例中则只包含译码器901和1个标签列903,且该标签列903中的每个表项对应一个索引号。因此,输入的索引号813先经译码器901译码得到字线(word line)后,再根据该字线选择标签列903中的相应表项的内容读出。此外,本实施例的标签部分还包括若干结构相同的独立标签模块(如图9中的独立标签模块905、907和909)。

[0089] 以独立标签模块905为例,包含索引号寄存器911、标签寄存器913、比较器915、传输门917和选择器919。其中,标签寄存器913中存储的内容与二维表825的表项内容相同,包含标签、有效位和组号;选择器919的输入来源于译码器901的输出字线;索引号寄存器911存储了该独立标签模块对应的索引号,该索引号被用做选择器919的控制信号,对相应字线进行选择。这样,只有当译码器901的输入索引号值等于索引号寄存器911存储的索引号值时,选择器919的输出为‘1’,否则输出为‘0’。该输出被用于控制标签寄存器913是否输出其中存储的标签到比较器915与存储器地址805中的标签811进行比较。具体地,若选择器919输出为‘1’,则表示独立标签模块905对应存储器地址805中的索引号813,在标签寄存器913中有效位为‘1’的情况下,存储的标签被送到比较器915与存储器地址805中的标签811进行比较,否则不进行比较。比较器915的比较结果被送到总线829上与比较器921及其他独立标签模块输出的比较结果进行“或”操作,得到缓存命中或缺失的结果。传输门917则与选择器923中的传输门类似,其输入为标签寄存器913中的组号,在比较器915输出比较结果为匹配的情况下,输出该组号819,与存储器地址805中的偏移地址815一同构成缓存寻址地址809。

[0090] 这样,除了可以按图8A实施例所述方法对标签列903进行配置,实现可重构缓存外,还可以对所述独立标签模块进行配置,将这些独立标签模块分配给相应的索引号对应的存储器地址使用,以达到更好的灵活性。

[0091] 例如,当标签列903中对应某个索引号的那个表项已经有效,且又发生基于该索引号对应的新存储器地址的指令或数据访问,若尚有未被占用的独立标签模块存在,则可以不对该标签列903中的该表项进行替换,而是使用独立标签模块存储该新地址中的标签,起到增加一个路(way)的效果。若所有独立标签模块也都被占用,则可以根据替换算法从这些独立标签模块中分配一个用于替换。这样,在本实施例中,可以利用这些独立标签模块,随着程序的运行实时分配路组,使得某些索引号能够对应超过一个路(way),而另一些索引号只对应一个路(way)或不对应路(way),以更合理有效地利用缓存,提高性能。

[0092] 根据本发明技术方案和构思,还可以有其他任何合适的改动。对于本领域普通技术人员来说,所有这些替换、调整和改进都应属于本发明所附权利要求的保护范围。

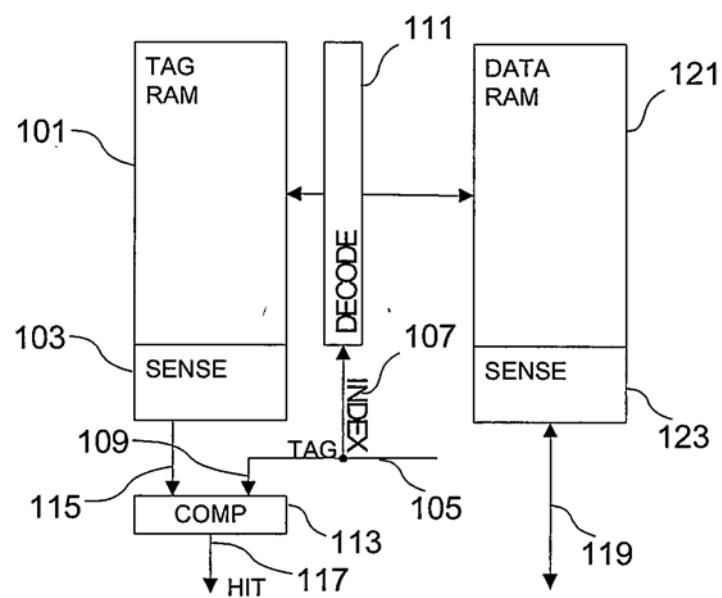


图1

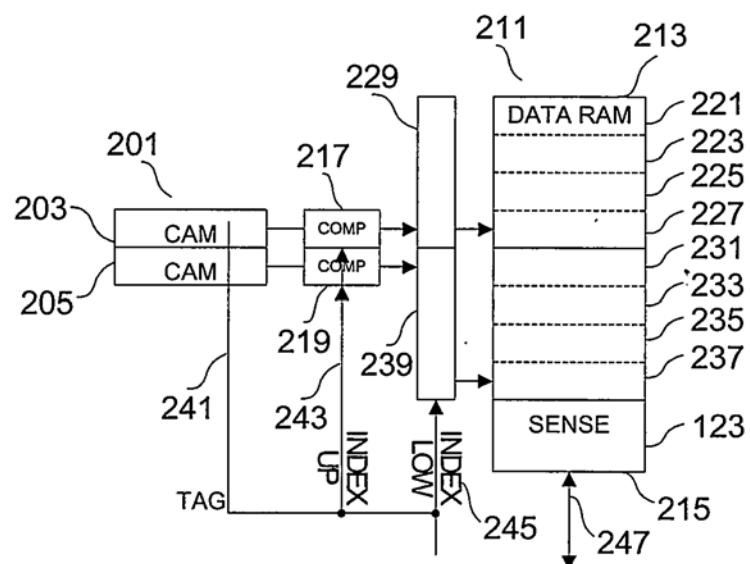


图2

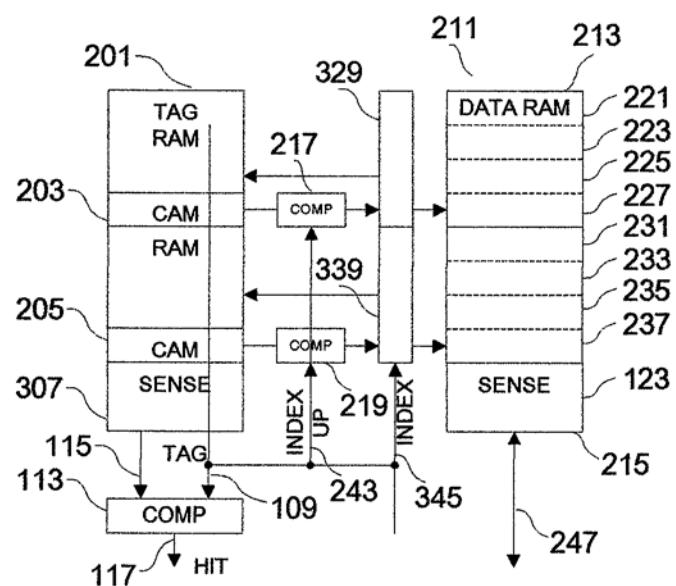


图3

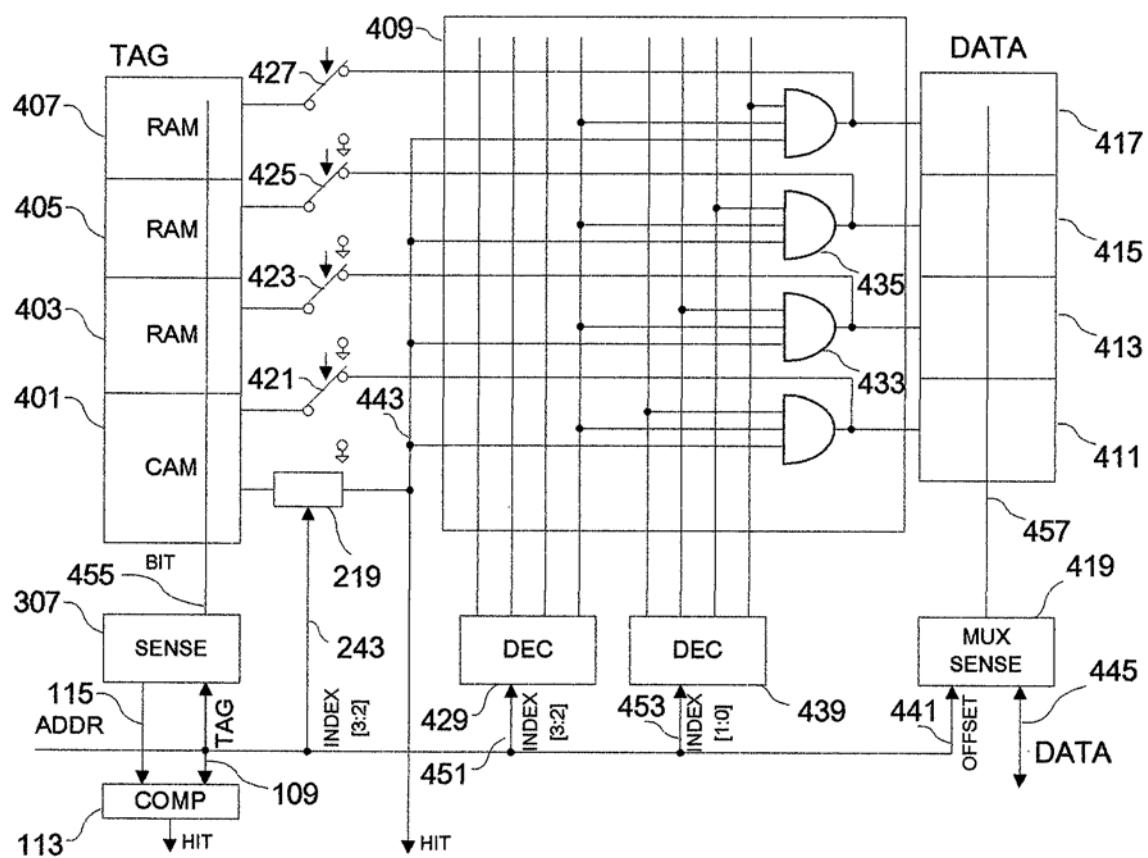


图4

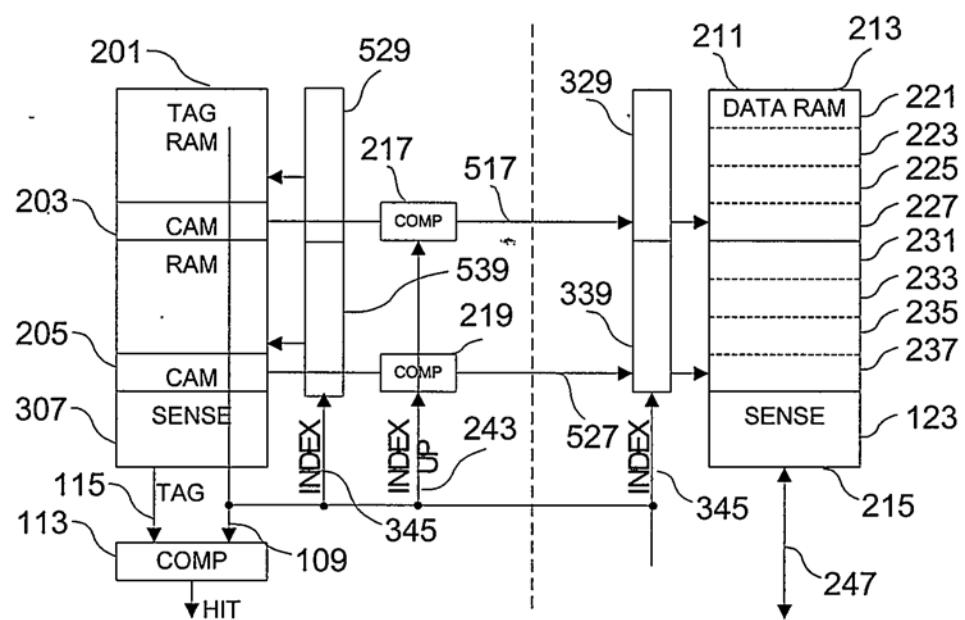


图5

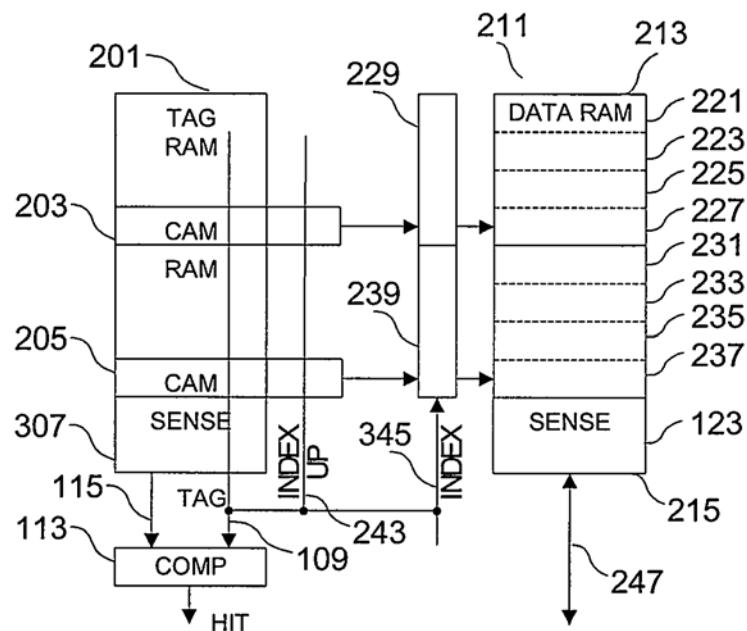


图6

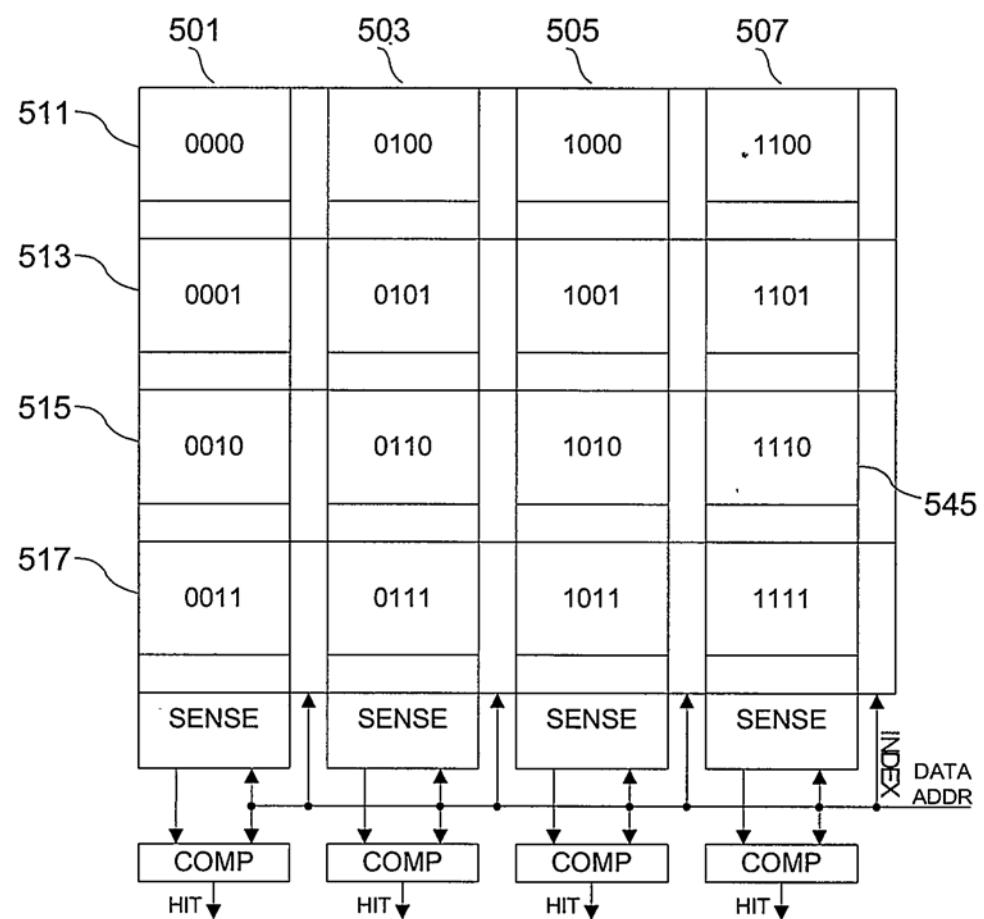


图7

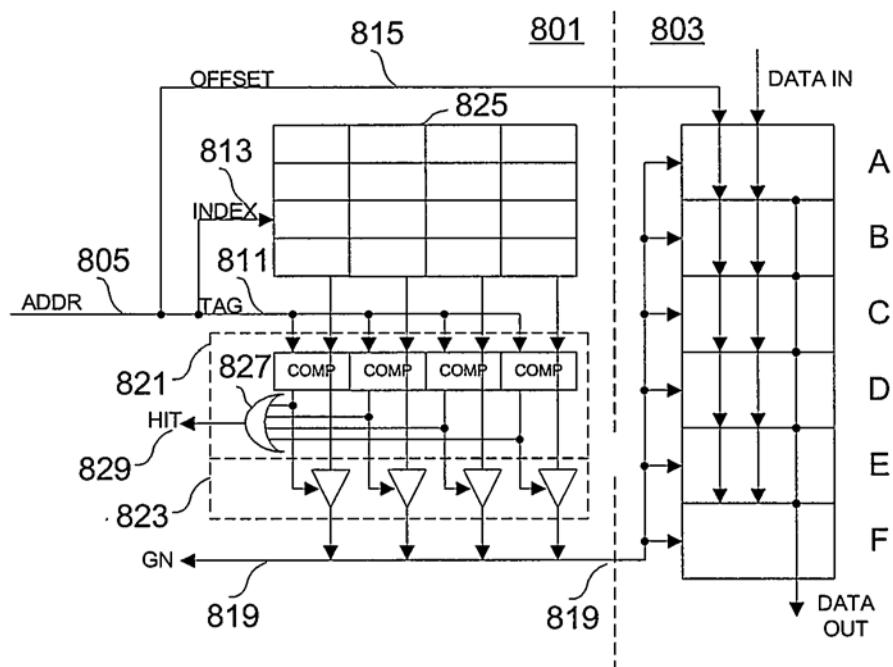


图8A

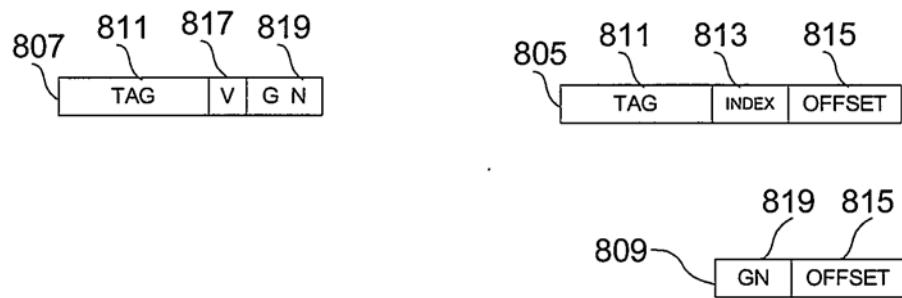


图8B

	0	1	2	3
3			F	
2		A		
1				
0	B	D	C	E

图8C

	0	1	2	3
3		C		
2	B			F
1		E	A	
0	D			

图8D

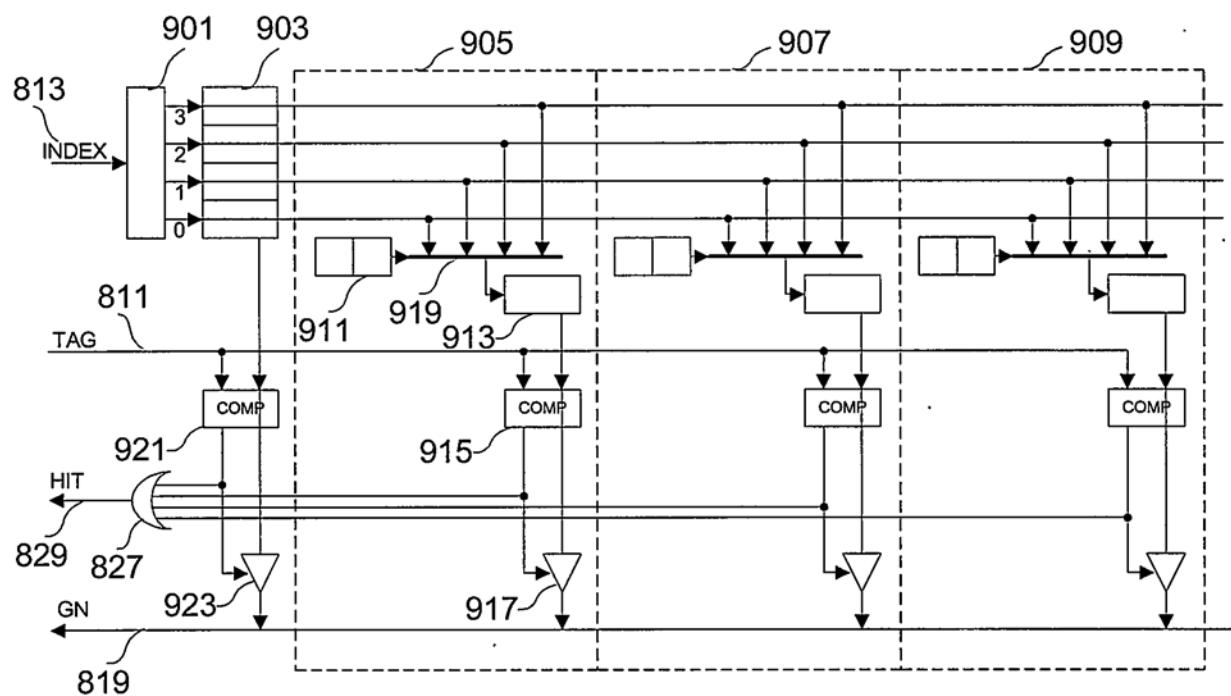


图9