



(12) 发明专利

(10) 授权公告号 CN 110780840 B

(45) 授权公告日 2023. 10. 31

(21) 申请号 201911047883.0

(22) 申请日 2019.10.30

(65) 同一申请的已公布的文献号
申请公布号 CN 110780840 A

(43) 申请公布日 2020.02.11

(73) 专利权人 湖南国科微电子股份有限公司
地址 410131 湖南省长沙市经济开发区泉塘街道东十路南段9号

(72) 发明人 魏世朋

(74) 专利代理机构 北京弘权知识产权代理有限公司 11363
专利代理师 逯长明 许伟群

(51) Int. Cl.
G06F 7/08 (2006.01)

(56) 对比文件
CN 104570144 A, 2015.04.29

CN 101192847 A, 2008.06.04

CN 1947091 A, 2007.04.11

US 2013086366 A1, 2013.04.04

US 5726923 A, 1998.03.10

CN 1987771 A, 2007.06.27

CN 101472055 A, 2009.07.01

CN 101620263 A, 2010.01.06

CN 104317549 A, 2015.01.28

CN 105512179 A, 2016.04.20

CN 108736861 A, 2018.11.02

CN 108763421 A, 2018.11.06

CN 109766074 A, 2019.05.17

US 2004148320 A1, 2004.07.29

杨绍岩; 刘万成; 周长春. 基于IRFPA改进中值滤波算法的FPGA实现. 光电技术应用. 2011, (第02期), 第64-70页.

审查员 余祖滢

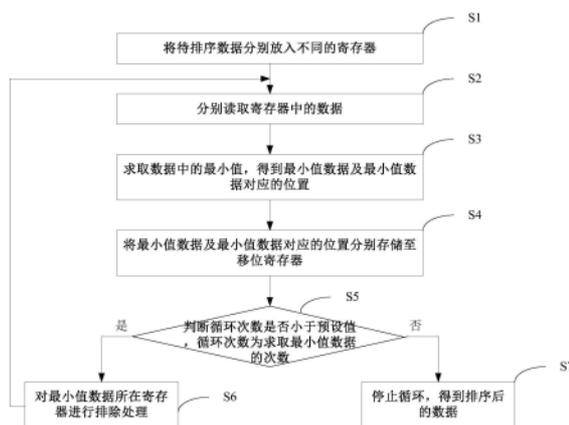
权利要求书2页 说明书6页 附图2页

(54) 发明名称

多路排序器的实现方法及系统

(57) 摘要

本申请公开了一种多路排序器的实现方法及系统,该方法包括:将待排序数据分别放入不同的寄存器;分别读取寄存器中的数据;求取数据中的最小值,得到最小值数据及最小值数据对应的位置;将最小值数据及最小值数据对应的位置分别存储至移位寄存器;判断循环次数是否小于预设值,其中,循环次数为求取最小值数据的次数;若循环次数小于预设值,则对最小值数据所在寄存器进行排除处理,继续求取数据中的最小值;若循环次数大于或等于预设值,则停止循环,得到排序后的数据。该方法采用获取最小值来代替实际的排序操作,每次获取的最小值由移位寄存器进行“移位寄存”,使得该方法耗费的逻辑资源少,很好的平衡了计算延时和逻辑资源,大大降低了成本。



CN 110780840 B

1. 一种多路排序器的实现方法,其特征在于,所述方法包括:
 - S1:将待排序数据分别放入不同的寄存器;
 - S2:分别读取所述寄存器中的数据;
 - S3:将读取的数据输入至最小值获取器中,通过所述最小值获取器求取所述数据中的最小值,得到最小值数据及最小值数据对应的位置;
 - S4:根据循环次数将所述最小值数据及最小值数据对应的位置分别存储至移位寄存器的相应位置;其中,所述循环次数为求取最小值数据的次数;
 - S5:判断循环次数是否小于预设值,若是,则执行步骤S6;若否,则执行步骤S7;
 - S6:对所述最小值数据所在寄存器进行排除处理,继续执行S2、S3、S4和S5;
 - S7:停止循环,得到排序后的数据。
2. 根据权利要求1所述的方法,其特征在于,根据循环次数将所述最小值数据及最小值数据对应的位置分别存储至移位寄存器的相应位置,包括:
 - 获取所述最小值数据对应的循环次数;
 - 根据所述循环次数将所述最小值数据和所述最小值数据对应的位置分别存储至第一移位寄存器和第二移位寄存器的相应位置。
3. 根据权利要求1所述的方法,其特征在于,对所述最小值数据所在寄存器进行排除处理,包括:
 - 将所述最小值数据所在寄存器内的数据替换为无穷大值。
4. 根据权利要求1所述的方法,其特征在于,对所述最小值数据所在寄存器进行排除处理,包括:
 - 将所述最小值数据所在寄存器内的数据删除。
5. 一种多路排序器的实现系统,其特征在于,包括:
 - 寄存器,用于存储待排序的数据;
 - 最小值获取模块,用于求取所述数据中的最小值,得到最小值数据及最小值数据对应的位置;
 - 移位寄存器,用于根据循环次数在相应位置分别存储所述最小值数据及所述最小值数据对应的位置;其中,所述循环次数为求取最小值数据的次数;
 - 判断模块,用于判断所述循环次数是否小于预设值;
 - 排除处理模块,用于当所述循环次数小于所述预设值时,对所述最小值数据所在寄存器进行排除处理;
 - 停止循环模块,用于当所述循环次数大于或等于所述预设值时,则停止循环,得到排序后的数据。
6. 根据权利要求5所述的系统,其特征在于,所述移位寄存器包括:
 - 第一移位寄存器,用于根据所述最小值数据对应的循环次数存储所述最小值数据;
 - 第二移位寄存器,用于根据所述最小值数据对应的循环次数存储所述最小值数据对应的位置。
7. 根据权利要求5所述的系统,其特征在于,所述排除处理模块包括:
 - 替换模块,用于将所述最小值数据所在寄存器内的数据替换为无穷大值。
8. 根据权利要求5所述的系统,其特征在于,所述排除处理模块包括:

删除模块,用于将所述最小值数据所在寄存器内的数据删除。

多路排序器的实现方法及系统

技术领域

[0001] 本申请涉及电子电路及半导体技术领域,尤其涉及一种多路排序器的实现方法及系统。

背景技术

[0002] 排序是使一串记录按照其中的某个或某些关键字的大小,递增或递减的排列起来的操作,该操作在自动控制、网页搜索、办公文档操作等大量使用。排序算法,就是如何使得记录按照要求排列的方法。

[0003] 经典的排序算法有冒泡排序、选择排序、插入排序等等,其中,冒泡排序是一种简单的排序算法,它重复地走访要排序的数列,一次比较两个元素,如果它们的顺序错误就把它们交换过来,走访数列的工作重复进行直到没有元素再需要交换,也就是说该数列已经排序完成;选择排序是表现最稳定的排序算法之一,其工作原理为:首先在未排序序列中找到最小(大)元素,存放到排序序列的起始位置,然后,再从剩余未排序元素中继续寻找最小(大)元素,然后放到已排序序列的末尾,以此类推,直到所有元素均排序完毕;插入排序是一种简单直观的排序算法,它的工作原理是:通过构建有序序列,对于未排序数据,在已排序序列中从后向前扫描,找到相应位置并插入。

[0004] 但是,这些排序算法大都适合采用软件方式实现,如果将这些算法用RTL硬件电路实现的话,就会产生处理延时与逻辑资源的矛盾,即计算延时小的排序算法耗费的逻辑资源非常大,耗费逻辑资源少的排序算法计算延迟又会很大,经典排序算法无法在逻辑资源和计算延时之间取得很好的平衡。

发明内容

[0005] 本申请提供了一种多路排序器的实现方法及系统,以解决目前经典排序算法无法在逻辑资源和计算延时之间取得很好平衡的问题。

[0006] 为了解决上述技术问题,本申请实施例公开了如下技术方案:

[0007] 第一方面,本申请实施例公开了一种多路排序器的实现方法,所述方法包括:

[0008] S1:将待排序数据分别放入不同的寄存器;

[0009] S2:分别读取所述寄存器中的数据;

[0010] S3:求取所述数据中的最小值,得到最小值数据及最小值数据对应的位置;

[0011] S4:将所述最小值数据及最小值数据对应的位置分别存储至移位寄存器;

[0012] S5:判断循环次数是否小于预设值,若是,则执行步骤S6;若否,则执行步骤S7;其中,所述循环次数为求取最小值数据的次数;

[0013] S6:对所述最小值数据所在寄存器进行排除处理,继续执行S2、S3、S4和S5;

[0014] S7:停止循环,得到排序后的数据。

[0015] 可选的,求取所述数据中的最小值,得到最小值数据及最小值数据对应的位置,包括:

- [0016] 获取所述寄存器中数据的最小值；
- [0017] 根据所述最小值数据获取所述最小值数据对应的位置。
- [0018] 可选的,将所述最小值数据及最小值数据对应的位置分别存储至移位寄存器,包括:
- [0019] 获取所述最小值数据对应的循环次数；
- [0020] 根据所述循环次数将所述最小值数据和所述最小值数据对应的位置分别存储至第一移位寄存器和第二移位寄存器。
- [0021] 可选的,对所述最小值数据所在寄存器进行排除处理,包括:
- [0022] 将所述最小值数据所在寄存器内的数据替换为无穷大值。
- [0023] 可选的,对所述最小值数据所在寄存器进行排除处理,包括:
- [0024] 将所述最小值数据所在寄存器内的数据删除。
- [0025] 第二方面,本申请实施例还公开了一种多路排序器的实现系统,包括:
- [0026] 寄存器,用于存储待排序的数据；
- [0027] 最小值获取模块,用于求取所述数据中的最小值,得到最小值数据及最小值数据对应的位置；
- [0028] 移位寄存器,用于分别存储所述最小值数据及所述最小值数据对应的位置；
- [0029] 判断模块,用于判断循环次数是否小于预设值,所述循环次数为求取最小值数据的次数；
- [0030] 排除处理模块,用于当所述循环次数小于所述预设值时,对所述最小值数据所在寄存器进行排除处理；
- [0031] 停止循环模块,用于当所述循环次数大于或等于所述预设值时,则停止循环,得到排序后的数据。
- [0032] 可选的,所述移位寄存器包括:
- [0033] 第一移位寄存器,用于根据所述最小值数据对应的循环次数存储所述最小值数据；
- [0034] 第二移位寄存器,用于根据所述最小值数据对应的循环次数存储所述最小值数据对应的位置。
- [0035] 可选的,所述排除处理模块包括:
- [0036] 替换模块,用于将所述最小值数据所在寄存器内的数据替换为无穷大值。
- [0037] 可选的,所述排除处理模块包括:
- [0038] 删除模块,用于将所述最小值数据所在寄存器内的数据删除。
- [0039] 与现有技术相比,本申请的有益效果为:
- [0040] 本申请实施例提供的多路排序器的实现方法包括:将待排序数据分别放入不同的寄存器;分别读取寄存器中的数据;求取数据中的最小值,得到最小值数据及最小值数据对应的位置;将最小值数据及最小值数据对应的位置分别存储至寄存器;判断循环次数是否小于预设值,循环次数为求取最小值数据的次数;若循环次数小于预设值,则对最小值数据所在寄存器进行排除处理,继续求取最小值数据等操作;若循环次数大于或等于预设值,则停止循环,得到排序后的数据。本申请提供的多路排序器的实现方法采用获取多路最小值来代替实际的排序操作,每次获取的最小值由移位寄存器进行“移位寄存”,使得该方法耗

费的逻辑资源少,逻辑处理延时适中,能够很好的平衡计算延时和逻辑资源,大大降低成本。

[0041] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本申请。

附图说明

[0042] 为了更清楚地说明本申请的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,显而易见地,对于本领域普通技术人员而言,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。

[0043] 图1为本申请实施例提供一种多路排序器的实现方法的流程图;

[0044] 图2为本申请实施例提供一种多路排序器的实现系统的框图;

[0045] 图3为本申请实施例提供的另一种多路排序器的实现系统的框图。

具体实施方式

[0046] 为了使本技术领域的人员更好地理解本申请中的技术方案,下面将结合本申请实施例中的附图,对本申请实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本申请一部分实施例,而不是全部实施例。基于本申请中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都应当属于本申请保护的范围。

[0047] 参见图1,为本申请实施例提供一种多路排序器的实现方法的流程图。

[0048] 如图1所示,本申请实施例提供的多路排序器的实现方法包括:

[0049] S1:将待排序数据分别放入不同的寄存器。

[0050] 本申请以64路排序器进行说明,将64个数据data1-data64分别暂存于64个寄存器之中。

[0051] S2:分别读取寄存器中的数据。

[0052] 根据需要读取寄存器中的数据,如需要对64个数据进行排序,则分别从64个寄存器中读取64个数据。

[0053] S3:求取数据中的最小值,得到最小值数据及最小值数据对应的位置。

[0054] 将读取的数据(64个数据)输入至最小值获取器中,得到64个数据中的最小值数据min_data及最小值数据对应的位置min_data_id,如最小值为2,它位于第15寄存器中。

[0055] S4:将最小值数据及最小值数据对应的位置分别存储至移位寄存器。

[0056] 得到64个数据中的最小值数据及最小值数据对应的位置后,将其分别存储至移位寄存器中,可根据排序要求将其存储至移位寄存器的首位或末位,如对64路数据由大到小排序,则将64路数据中的最小值数据及最小值数据对应的位置分别存储至移位寄存器的末位;对64路数据由小到大排序,则将64路数据中的最小值数据及最小值数据对应的位置分别存储至移位寄存器的首位。

[0057] 本示例中,可通过最小值数据对应的循环次数将最小值数据与最小值数据对应的位置分别存储至第一移位寄存器和第二移位寄存器,如本次得到最小值数据对应的循环次数为2,即第二次求取数据的最小值,若对64路数据由大到小排序,则将最小值数据存储至

第一移位寄存器的右边第2位,将最小值数据对应的位置存储至第二移位寄存器的右边第2位;若对64路数据由小到大排序,则将最小值数据存储至第一移位寄存器的左边第2位,将最小值数据对应的位置存储至第二移位寄存器的左边第2位。

[0058] S5:判断循环次数是否小于预设值,若是,则执行步骤S6;若否,则执行步骤S7;其中,循环次数为求取最小值数据的次数。

[0059] 每次得到最小值数据及最小值数据对应的位置,则循环次数加1,若要对64个数据进行排序,则设置预设值为64,判断循环次数是否小于64;若不需要排序这么多数据,而是只需要挑选一部分求取最小值,如24个,则只需将预设值设定为24即可。若循环次数小于预设值,则执行S6;若循环次数大于或等于预设值,则执行S7。

[0060] S6:对最小值数据所在寄存器进行排除处理,继续执行S2、S3、S4和S5。

[0061] 若求取最小值数据的循环次数小于预设值,如对64路数据第一次求取最小值,即循环次数小于预设值64,则对最小值数据所在寄存器进行排除处理,继续执行S2、S3、S4和S5,即再次读取寄存器中的数据,求取数据中的最小值,得到最小值数据及最小值数据对应的位置,将最小值数据及最小值数据对应的位置分别存储至第一移位寄存器与第二移位寄存器,判断循环次数是否小于预设值。

[0062] 对最小值数据所在寄存器进行排除处理时,可将该最小值数据所在寄存器内的数据替换为无穷大值。如将第一次求取最小值数据所在寄存器内的数据替换为无穷大值,之后读取寄存器中包含无穷大值的64路数据,再次求取64路数据中的最小值,得到第二次的最小值数据及最小值数据对应的位置,并根据排序要求将第二次的最小值数据及最小值数据对应的位置分别存储至第一移位寄存器与第二移位寄存器的第二位或倒数第二位。

[0063] 对最小值数据所在寄存器进行排除处理时,也可将最小值数据所在寄存器内的数据删除。如删除第一次求取最小值数据所在寄存器内的数据,则只有63个寄存器内有数据,之后读取63个寄存器中的数据,求取63路数据中的最小值,得到第二次的最小值数据及最小值数据对应的位置,并根据排序要求将第二次的最小值数据及最小值数据对应的位置分别存储至第一移位寄存器与第二移位寄存器的第二位或倒数第二位。

[0064] 按照上述方法对S2、S3、S4、S5和S6进行64次循环,得到64个最小值数据及最小值数据对应的位置,并根据排序要求将64个最小值数据及最小值数据分别存储至第一移位寄存器与第二移位寄存器。

[0065] 本申请获取64次最小值数据及最小值数据对应位置的方法并不仅限于上述方法,只要能循环得到每次的最小值数据及最小值数据对应的位置即可,其均属于本申请实施例的保护范围。

[0066] S7:停止循环,得到排序后的数据。

[0067] 如果获取最小值数据的循环次数大于或等于预设值,则停止循环,此时移位寄存器中的内容即为排序后的数据,如此完成64路数据的排序。

[0068] 本申请实施例提供的多路排序器的实现方法采用获取64路最小值来代替实际的排序操作,每次只获取64路数据中的最小值,每次获取的最小值由移位寄存器进行“移位寄存”,且每次获取最小值后要及时修改寄存器对应位置的内容(用无穷大代替或删除对应寄存器内的数据),如此耗费的逻辑资源少,只需要64个寄存器、1个64路最小值获取器、2排移位寄存器即可,逻辑处理延时适中,很好的平衡了计算延时和逻辑资源,结构新颖简单,节

约面积,大大降低了成本。

[0069] 基于上述实施例所述的多路排序器的实现方法,本申请实施例还提供了一种多路排序器的实现系统。

[0070] 如图2、图3所示,本申请实施例提供的多路排序器的实现系统包括:

[0071] 寄存器,用于存储待排序的数据。本示例中,需要64个寄存器,分别暂存64个数据data1-data64。

[0072] 最小值获取模块,用于求取数据中的最小值,得到最小值数据及最小值数据对应的位置。

[0073] 移位寄存器,用于分别存储最小值数据及最小值数据对应的位置。本示例中,移位寄存器包括第一移位寄存器与第二移位寄存器,其中,

[0074] 第一移位寄存器,用于根据最小值数据对应的循环次数存储最小值数据。本示例中,获取最小值数据对应的循环次数后,根据循环次数将最小值数据存储至第一移位寄存器中,如将64路数据由小到大排序时,按照循环次数将最小值数据存储至第一移位寄存器的左边相应位置。如将64路数据由大到小排序时,按照循环次数将最小值数据存储至第一移位寄存器的右边相应位置。

[0075] 第二移位寄存器,用于根据最小值数据对应的循环次数存储最小值数据对应的位置。本示例中,获取最小值数据对应的循环次数后,根据循环次数将最小值数据对应的位置存储至第二移位寄存器中,如将64路数据由小到大排序时,按照循环次数将最小值数据对应的位置存储至第二移位寄存器的左边相应位置。如将64路数据由大到小排序时,按照循环次数将最小值数据对应的位置存储至第二移位寄存器的右边相应位置。

[0076] 判断模块,用于判断循环次数是否小于预设值,其中,循环次数为求取最小值数据的次数。

[0077] 排除处理模块,用于当循环次数小于预设值时,对最小值数据所在寄存器进行排除处理。排除处理模块可包括替换模块,也可包括删除模块。

[0078] 替换模块,用于将最小值数据所在寄存器内的数据替换为无穷大值,即根据求取的最小值数据将最小值数据所在寄存器内的数据替换为无穷大值,之后读取替换后的寄存器中的数据,重新求取数据中的最小值,得到最小值数据及最小值数据对应的位置,并将最小值数据及最小值数据对应的位置分别存储至第一移位寄存器与第二移位寄存器。

[0079] 删除模块,用于将最小值数据所在寄存器内的数据删除,即根据求取的最小值数据将最小值数据所在寄存器内的数据删除,之后读取剩余寄存器中的数据,重新求取数据中的最小值,得到最小值数据及最小值数据对应的位置,并将最小值数据及最小值数据对应的位置分别存储至第一移位寄存器与第二移位寄存器。

[0080] 本申请实施例提供的多路排序器的实现系统耗费的逻辑资源少,只需要多个寄存器、1个最小值获取器、2排移位寄存器即可,逻辑处理延时适中,能够很好的平衡计算延时和逻辑资源,大大降低成本。

[0081] 本领域技术人员在考虑说明书及实践这里发明的公开后,将容易想到本申请的其他实施方案。本申请旨在涵盖本发明的任何变型、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申请的一般性原理并包括本申请未公开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,本申请的真正范围和精神由权利要

求的内容指出。

[0082] 以上所述的本申请实施方式并不构成对本申请保护范围的限定。

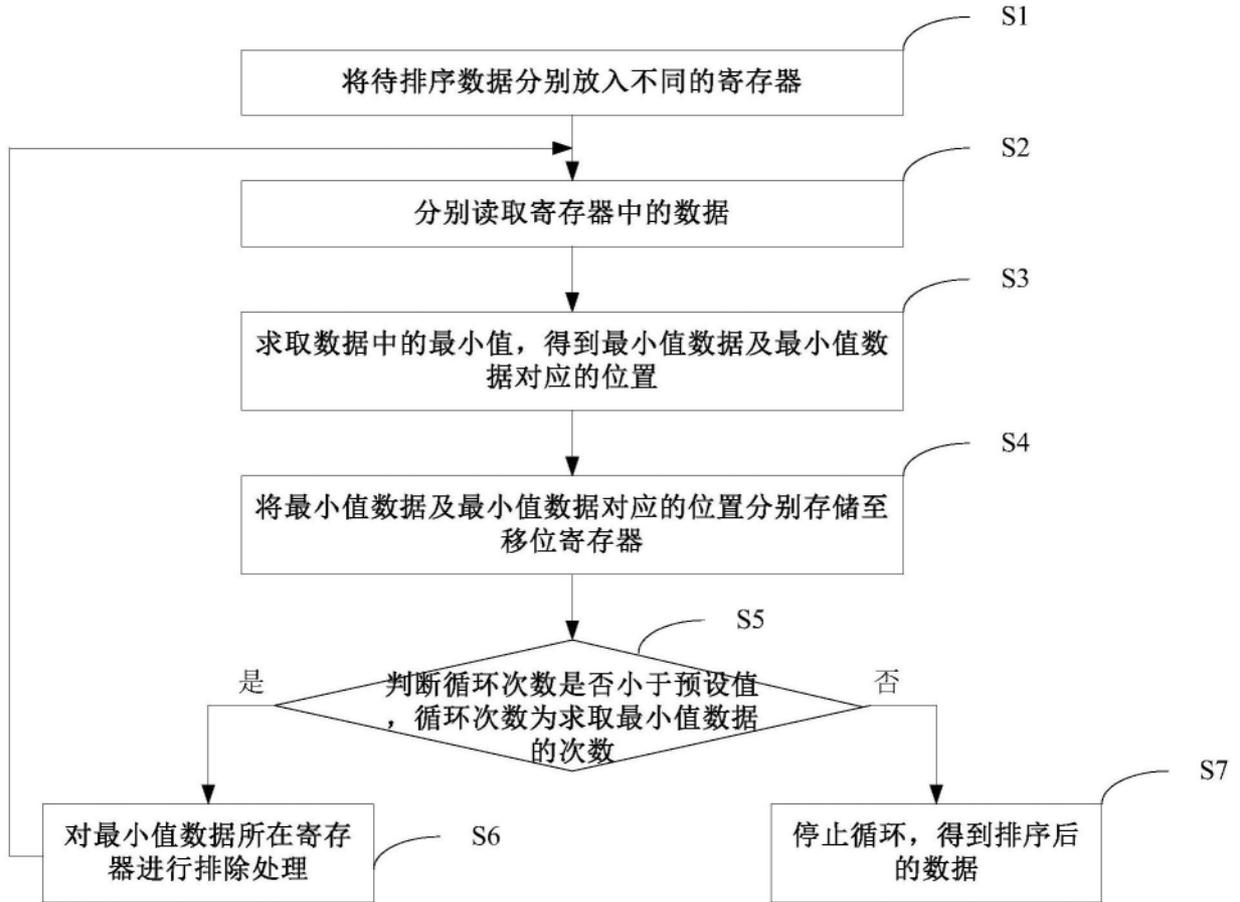


图1

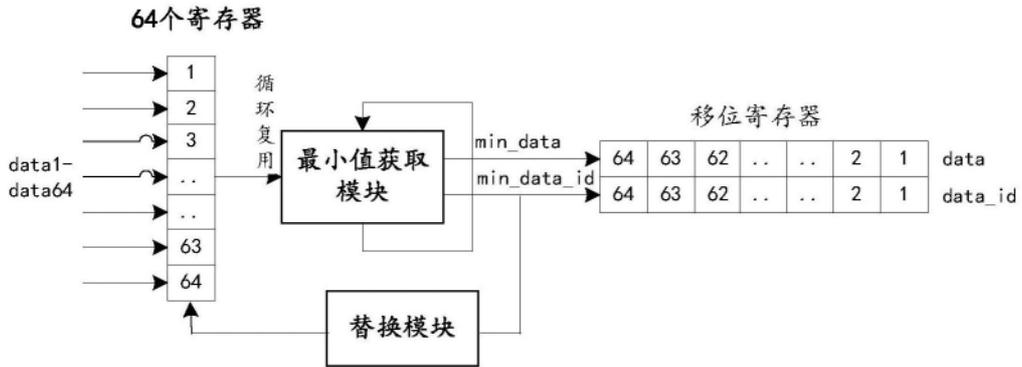


图2

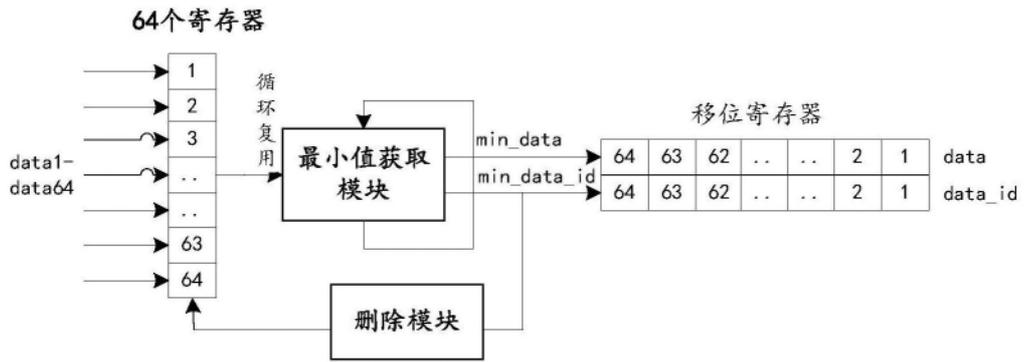


图3