



(12) 发明专利申请

(10) 申请公布号 CN 101833511 A

(43) 申请公布日 2010.09.15

(21) 申请号 201010136562.0

(22) 申请日 2010.03.29

(71) 申请人 瑞斯康达科技发展股份有限公司

地址 100085 北京市海淀区上地信息路 11
号彩虹大厦南楼 3 层 323

(72) 发明人 王夕军 宋显建

(74) 专利代理机构 北京安信方达知识产权代理
有限公司 11262

代理人 栗若木 王漪

(51) Int. Cl.

G06F 12/02 (2006.01)

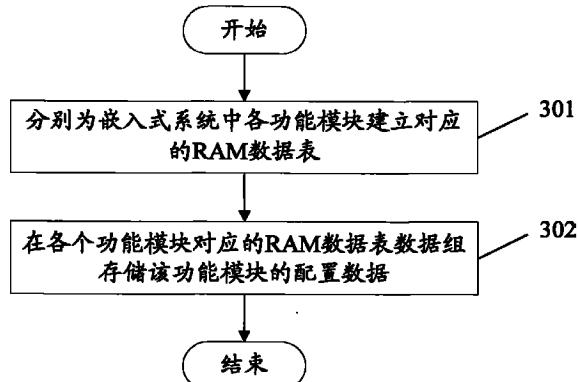
权利要求书 3 页 说明书 12 页 附图 14 页

(54) 发明名称

数据管理方法、装置和系统

(57) 摘要

本发明提供了一种数据管理方法、装置和系统。涉及嵌入式系统；解决了传统配置数据管理方式效率低实现复杂的问题。该方法包括：分别为嵌入式系统中各功能模块建立对应的 RAM 数据表；在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据。本发明提供的技术方案适用于嵌入式系统中。



1. 一种数据管理方法,其特征在于,包括:

分别为嵌入式系统中各功能模块建立对应的 RAM 数据表;

在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据。

2. 根据权利要求 1 所述的数据管理方法,其特征在于,所述在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据的步骤包括:

创建包含所述配置数据的待存储的记录列表,每个记录列表均包含一个主键;

当所述 RAM 数据表中不存在与所述待存储的记录列表中所包含的主键字段相同的记录列表,则在该 RAM 数据表的尾部添加所述待存储的记录列表。

3. 根据权利要求 2 所述的数据管理方法,其特征在于,所述在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据的步骤还包括:

当所述 RAM 数据表中存在与所述待存储的记录列表中所包含的主键字段相同的记录列表,则不添加所述待存储的记录列表。

4. 根据权利要求 2 所述的数据管理方法,其特征在于,所述 RAM 数据表包括至少一个索引,每个索引对应一个正向索引表,所述正向索引表按照该组索引的顺序存储该 RAM 数据表中各记录列表的记录编号,所述在该 RAM 数据表的尾部添加所述待存储的记录列表的步骤之后,还包括:

使用快速排序算法对各组索引的正向索引表进行重新排序。

5. 根据权利要求 4 所述的数据管理方法,其特征在于,每个索引还对应一个反向索引表,所述反向索引表存储该 RAM 数据表中各记录列表的按该组索引的顺序排序后的记录名次,所述使用快速排序算法对各组索引的正向索引表进行重新排序的步骤之后,还包括:

各组索引的反向索引表根据对应的正向索引表,更改其中存储的记录名次。

6. 根据权利要求 1 所述的数据管理方法,其特征在于,还包括:

接收检索记录指令,该指令中包含至少一个检索条件;

根据各检索条件分别进行检索,获取各检索条件的检索结果集合,所述检索条件具体为一个字段的取值范围或指定值;

取各检索结果集合的交集作为检索记录的检索结果。

7. 根据权利要求 6 所述的数据管理方法,其特征在于,所述取各检索结果集合的交集作为检索记录的检索结果的步骤包括:

分别为各个检索结果集合构建一个位图,该位图的第 k bit 对应 RAM 数据表中记录编号为 k 的记录列表;

将后一检索结果集合累加到其之前的检索结果集合交集中,得到交集的位图,将该位图中交集内的记录列表对应的 bit 值置为 1;

在取完全部检索结果集合的交集后,将最终获取的位图中 bit 值为 1 的 bit 对应的记录列表的集合作为检索结果。

8. 根据权利要求 5 所述的数据管理方法,其特征在于,该方法还包括:

接收修改指令,所述修改指令中包含至少一个检索条件;

根据所述检索条件,检索到相应的记录列表;

对所述记录列表进行修改。

9. 根据权利要求 5 所述的数据管理方法,其特征在于,该方法还包括:

接收删除指令,所述删除指令中包含至少一个检索条件;
根据所述检索条件,检索到相应的记录列表;
删除所述记录列表。

10. 根据权利要求 5 所述的数据管理方法,其特征在于,该方法还包括:

接收检索下一个记录指令,该指令中包含至少一个检索条件;
根据各检索条件依次进行检索,分别获取各有效检索条件的检索结果集合;
取各检索结果集合的交集;
查询最后一个有效检索条件对应的一个索引中的反向索引表,获取所述交集中的各记录列表的记录名次;
将记录名次最小的记录列表作为检索下一个记录的检索结果。

11. 根据权利要求 1 所述的数据管理方法,其特征在于,还包括:

建立 NVRAM 数据库,在所述 NVRAM 数据库中分别为嵌入式系统中各功能模块建立对应的 NVRAM 数据表,所述 NVRAM 存储数据表的当前记录和容量;
通过直接读写裸介质在所述 NVRAM 数据库与嵌入式系统中各功能模块对应的 RAM 数据表之间进行记录列表同步。

12. 根据权利要求 11 所述的数据管理方法,其特征在于,所述通过直接读写裸介质在所述 NVRAM 数据库与嵌入式系统中各功能模块对应的 RAM 数据表之间进行记录列表同步的步骤包括:

采用直接读取裸介质的方式将所述 NVRAM 数据表中的记录列表读取到内存中,加入到对应的 RAM 数据表内,并删除所述 NVRAM 中存储的记录列表;或,

检查 RAM 数据表内各记录列表的同步标志,将该同步标志为未同步状态的记录列表写入对应的 NVRAM 数据表内,并将所述记录列表的同步标志改为已同步状态。

13. 一种数据管理装置,其特征在于,包括:

RAM 数据表建立模块,用于分别为嵌入式系统中各功能模块建立对应的 RAM 数据表;
数据存储模块,用于在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据。

14. 根据权利要求 13 所述的数据管理装置,其特征在于,还包括:

检索记录模块,用于接收检索记录指令,该指令中包含至少一个检索条件,根据各检索条件分别对所述数据存储模块存储的 RAM 数据表数据组进行检索,获取各检索条件的检索结果集合,所述检索条件具体为一个字段的取值范围或指定值,取各检索结果集合的交集作为检索记录的检索结果。

15. 根据权利要求 13 所述的数据管理装置,其特征在于,还包括:

检索下一个记录模块,用于接收检索下一个记录指令,该指令中包含至少一个检索条件,根据所述检索条件依次对所述数据存储模块存储的 RAM 数据表数据组进行检索,分别获取各有效检索条件的检索结果集合,查询最后一个有效检索条件对应的一个索引中的反向索引表,获取所述检索结果集合中的各记录列表的记录名次,并将记录名次最小的记录列表作为检索下一个记录的检索结果。

16. 根据权利要求 13 所述的数据管理装置,其特征在于,还包括:

修改模块,用于接收修改指令,所述修改指令中包含至少一个检索条件,根据所述检索

条件,检索所述数据存储模块存储的 RAM 数据表数据组获取相应的记录列表,对所述记录列表进行修改。

17. 根据权利要求 13 所述的数据管理装置,其特征在于,还包括:

删除模块,用于接收删除指令,所述删除指令中包含至少一个检索条件,根据所述检索条件,检索所述数据存储模块存储的 RAM 数据表数据组获取相应的记录列表,对所述记录列表进行修改。

18. 一种数据管理系统,其特征在于,包括数据管理装置;

所述数据管理装置,用于分别为嵌入式系统中各功能模块建立对应的 RAM 数据表,并在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据。

19. 根据权利要求 18 所述的数据管理系统,其特征在于,还包括 NVRAM 数据库;

所述 NVRAM 数据库,用于分别为嵌入式系统中各功能模块建立对应的 NVRAM 数据表,并通过读取裸介质在所述 NVRAM 数据库与嵌入式系统中各功能模块对应的 RAM 数据表之间进行记录列表同步。

20. 根据权利要求 18 所述的数据管理系统,其特征在于,

所述数据管理装置,还用于接收检索记录指令,该指令中包含至少一个检索条件,根据各检索条件分别进行检索,获取各检索条件的检索结果集合,所述检索条件具体为一个字段的取值范围或指定值,取各检索结果集合的交集作为检索记录的检索结果;和 / 或

接收检索下一个记录指令,该指令中包含至少一个检索条件,根据所述检索条件依次进行检索,分别获取各有效检索条件的检索结果集合,查询最后一个有效检索条件对应的一个索引中的反向索引表,获取所述检索结果集合中的各记录列表的记录名次,并将记录名次最小的记录列表作为检索下一个记录的检索结果;和 / 或

接收修改指令,所述修改指令中包含至少一个检索条件,根据所述检索条件,检索到相应的记录列表,对所述记录列表进行修改;和 / 或

接收删除指令,所述删除指令中包含至少一个检索条件,根据所述检索条件,检索到相应的记录列表,对所述记录列表进行修改。

数据管理方法、装置和系统

技术领域

[0001] 本发明涉及嵌入式系统，尤其涉及一种嵌入式系统下的数据管理方法、装置和系统。

背景技术

[0002] 嵌入式通信设备系统中，存在较多的数据操作（保存、修改、查询），这些数据一般需要用户配置，统称为配置数据。系统运行过程中，配置数据一般保存在内存（以下缩写为 RAM）中，有的设备还需要存储于非易失性介质（以下缩写为 NVRAM）中。

[0003] 现有常用的数据库管理系统都比较大（一般都在兆级以上）。受硬件资源（CPU 性能、RAM/NVRAM 大小）限制，嵌入式通信系统无法采用这些数据库系统来管理配置数据。目前传统的配置数据管理方式是：在内存中，配置数据则分散保存于各功能模块内部，格式也由各功能模块自行定义；在需要存储到 NVRAM 时，再由各功能模块将其负责的配置数据转换为文本格式，存储到 NVRAM 中。

[0004] 随着实时通信系统功能的增强，系统配置数据的数量也在迅速增长。此时，各功能模块在管理各自负责的配置数据时都面临着同样的时间效率问题，都要根据各自的配置数据保存格式去分别优化改进，而文本格式的存储方式在配置数据数量较大时存储空间效率问题也暴露了出来。这使得传统配置数据管理方式在配置数据规模大大增长后基本不可用。

发明内容

[0005] 本发明提供了一种数据管理方法、装置和系统，解决了上述传统配置数据管理方式效率低实现复杂的问题。

[0006] 一种数据管理方法，包括：

[0007] 分别为嵌入式系统中各功能模块建立对应的 RAM 数据表；

[0008] 在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据。

[0009] 进一步的，所述在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据的步骤包括：

[0010] 创建包含所述配置数据的待存储的记录列表，每个记录列表均包含一个主键；

[0011] 当所述 RAM 数据表中不存在与所述待存储的记录列表中所包含的主键字段相同的记录列表，则在该 RAM 数据表的尾部添加所述待存储的记录列表。

[0012] 进一步的，所述在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据的步骤还包括：

[0013] 当所述 RAM 数据表中存在与所述待存储的记录列表中所包含的主键字段相同的记录列表，则不添加所述待存储的记录列表。

[0014] 进一步的，所述 RAM 数据表包括至少一个索引，每个索引对应一个正向索引表，所述正向索引表按照该组索引的顺序存储该 RAM 数据表中各记录列表的记录编号，所述在该

RAM 数据表的尾部添加所述待存储的记录列表的步骤之后,还包括:

- [0015] 使用快速排序算法对各组索引的正向索引表进行重新排序。
- [0016] 进一步的,每个索引还对应一个反向索引表,所述反向索引表存储该 RAM 数据表中各记录列表的按该组索引的顺序排序后的记录名次,所述使用快速排序算法对各组索引的正向索引表进行重新排序的步骤之后,还包括:
 - [0017] 各组索引的反向索引表根据对应的正向索引表,更改其中存储的记录名次。
 - [0018] 进一步的,上述数据管理方法还包括:
 - [0019] 接收检索记录指令,该指令中包含至少一个检索条件;
 - [0020] 根据各检索条件分别进行检索,获取各检索条件的检索结果集合,所述检索条件具体为一个字段的取值范围或指定值;
 - [0021] 取各检索结果集合的交集作为检索记录的检索结果。
 - [0022] 进一步的,所述取各检索结果集合的交集作为检索记录的检索结果的步骤包括:
 - [0023] 分别为各个检索结果集合构建一个位图,该位图的第 k bit 对应 RAM 数据表中记录编号为 k 的记录列表;
 - [0024] 将后一检索结果集合累加到其之前的检索结果集合交集中,得到交集的位图,将该位图中交集内的记录列表对应的 bit 值置为 1;
 - [0025] 在取完全部检索结果集合的交集后,将最终获取的位图中 bit 值为 1 的 bit 对应的记录列表的集合作为检索结果。
 - [0026] 进一步的,上述数据管理方法还包括:
 - [0027] 接收修改指令,所述修改指令中包含至少一个检索条件;
 - [0028] 根据所述检索条件,检索到相应的记录列表;
 - [0029] 对所述记录列表进行修改。
 - [0030] 进一步的,上述数据管理方法还包括:
 - [0031] 接收删除指令,所述删除指令中包含至少一个检索条件;
 - [0032] 根据所述检索条件,检索到相应的记录列表;
 - [0033] 删除所述记录列表。
 - [0034] 进一步的,上述数据管理方法还包括:
 - [0035] 接收检索下一个记录指令,该指令中包含至少一个检索条件;
 - [0036] 根据各检索条件依次进行检索,分别获取各有效检索条件的检索结果集合;
 - [0037] 取各检索结果集合的交集;
 - [0038] 查询最后一个有效检索条件对应的一个索引中的反向索引表,获取所述交集中的各记录列表的记录名次;
 - [0039] 将记录名次最小的记录列表作为检索下一个记录的检索结果。
 - [0040] 进一步的,上述数据管理方法还包括:
 - [0041] 建立 NVRAM 数据库,在所述 NVRAM 数据库中分别为嵌入式系统中各功能模块建立对应的 NVRAM 数据表,所述 NVRAM 存储数据表的当前记录和容量;
 - [0042] 通过直接读写裸介质在所述 NVRAM 数据库与嵌入式系统中各功能模块对应的 RAM 数据表之间进行记录列表同步。
 - [0043] 进一步的,所述通过直接读写裸介质在所述 NVRAM 数据库与嵌入式系统中各功能

模块对应的 RAM 数据表之间进行记录列表同步的步骤包括：

[0044] 采用直接读取裸介质的方式将所述 NVRAM 数据表中的记录列表读取到内存中，加入到对应的 RAM 数据表内，并删除所述 NVRAM 中存储的记录列表；或，

[0045] 检查 RAM 数据表内各记录列表的同步标志，将该同步标志为未同步状态的记录列表写入对应的 NVRAM 数据表内，并将所述记录列表的同步标志改为已同步状态。

[0046] 本发明还提供了一种数据管理装置，包括：

[0047] RAM 数据表建立模块，用于分别为嵌入式系统中各功能模块建立对应的 RAM 数据表；

[0048] 数据存储模块，用于在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据。

[0049] 进一步的，上述数据管理装置还包括：

[0050] 检索记录模块，用于接收检索记录指令，该指令中包含至少一个检索条件，根据各检索条件分别对所述数据存储模块存储的 RAM 数据表数据组进行检索，获取各检索条件的检索结果集合，所述检索条件具体为一个字段的取值范围或指定值，取各检索结果集合的交集作为检索记录的检索结果。

[0051] 进一步的，上述数据管理装置还包括：

[0052] 检索下一个记录模块，用于接收检索下一个记录指令，该指令中包含至少一个检索条件，根据所述检索条件依次对所述数据存储模块存储的 RAM 数据表数据组进行检索，分别获取各有效检索条件的检索结果集合，查询最后一个有效检索条件对应的一个索引中的反向索引表，获取所述检索结果集合中的各记录列表的记录名次，并将记录名次最小的记录列表作为检索下一个记录的检索结果。

[0053] 进一步的，上述数据管理装置还包括：

[0054] 修改模块，用于接收修改指令，所述修改指令中包含至少一个检索条件，根据所述检索条件，检索所述数据存储模块存储的 RAM 数据表数据组获取相应的记录列表，对所述记录列表进行修改。

[0055] 进一步的，上述数据管理装置还包括：

[0056] 删除模块，用于接收删除指令，所述删除指令中包含至少一个检索条件，根据所述检索条件，检索所述数据存储模块存储的 RAM 数据表数据组获取相应的记录列表，对所述记录列表进行修改。

[0057] 本发明还提供了一种数据管理系统，包括数据管理装置；

[0058] 所述数据管理装置，用于分别为嵌入式系统中各功能模块建立对应的 RAM 数据表，并在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据。

[0059] 进一步的，上述数据管理系统，还包括 NVRAM 数据库；

[0060] 所述 NVRAM 数据库，用于分别为嵌入式系统中各功能模块建立对应的 NVRAM 数据表，并通过读取裸介质在所述 NVRAM 数据库与嵌入式系统中各功能模块对应的 RAM 数据表之间进行记录列表同步。

[0061] 进一步的，所述数据管理装置，还用于接收检索记录指令，该指令中包含至少一个检索条件，根据各检索条件分别进行检索，获取各检索条件的检索结果集合，所述检索条件具体为一个字段的取值范围或指定值，取各检索结果集合的交集作为检索记录的检索结

果 ; 和 / 或

[0062] 接收检索下一个记录指令，该指令中包含至少一个检索条件，根据所述检索条件依次进行检索，分别获取各有效检索条件的检索结果集合，查询最后一个有效检索条件对应的一个索引中的反向索引表，获取所述检索结果集合中的各记录列表的记录名次，并将记录名次最小的记录列表作为检索下一个记录的检索结果；和 / 或

[0063] 接收修改指令，所述修改指令中包含至少一个检索条件，根据所述检索条件，检索到相应的记录列表，对所述记录列表进行修改；和 / 或

[0064] 接收删除指令，所述删除指令中包含至少一个检索条件，根据所述检索条件，检索到相应的记录列表，对所述记录列表进行修改。

[0065] 本发明的实施例提供了一种数据管理方法、装置和系统，在 RAM 数据库中分别为嵌入式系统中各功能模块建立对应的 RAM 数据表，在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据，实现了对嵌入式系统中各个功能块配置数据格式的统一管理，使得数据库结构更简单，简化了实现方式，提高了数据存储效率。使用位图对多个检索结果集合取交集，实现了取交集的线性效率。通过关系数据表格式存储的 NVRAM 数据库，也提高了配置数据存储空间的效率，在这一过程中还使用了直接读写 NVRAM 裸介质的方式，缩短了访问时间，降低了系统在写 NVRAM 期间所占用的系统资源。此外，通过使用反向索引表，也实现了嵌入式系统数据库的使 RAM 数据库“检索下一个”的功能，为所有功能模块解决了 SNMP 中对配置数据的“检索下一个”功能的时间效率问题。

附图说明

- [0066] 图 1 为本发明的实施例所使用的 RAM 数据库的逻辑结构示意图；
- [0067] 图 2 为本发明的实施例所使用的记录列表的结构示意图；
- [0068] 图 3 为本发明的实施例一提供的一种数据管理方法的流程图；
- [0069] 图 4 为图 3 中步骤 302 的具体流程图；
- [0070] 图 5 为本发明的实施例二提供的一种数据管理方法的流程图；
- [0071] 图 6 为图 5 中步骤 502 的具体流程图；
- [0072] 图 7 为图 5 中步骤 503 的具体流程图；
- [0073] 图 8 为图 5 中步骤 504 的具体流程图；
- [0074] 图 9 为图 5 中步骤 505 的具体流程图；
- [0075] 图 10 为本发明的实施例三提供的一种数据管理方法的流程图；
- [0076] 图 11 为本发明的实施例三中获取各有效检索条件检索集合的具体流程图；
- [0077] 图 12 为本发明的实施例三中获取各有效检索条件检索集合交集的具体流程图；
- [0078] 图 13 为图 10 中步骤 1003 的具体流程图；
- [0079] 图 14 为本发明的实施例四提供的一种数据管理方法的流程图；
- [0080] 图 15 为本发明的实施例使用的 NVRAM 数据库结构示意图；
- [0081] 图 16 为本发明的实施例提供的一种数据管理装置的结构示意图；
- [0082] 图 17 为本发明的又一实施例提供的一种数据管理装置的结构示意图。

具体实施方式

[0083] 随着实时通信系统功能的增强,系统配置数据的数量也在迅速增长。此时,各功能模块在管理各自负责的配置数据时都面临着同样的时间效率问题,都要根据各自的配置数据保存格式去分别优化改进,而文本格式的存储方式在配置数据数量较大时存储空间效率问题也暴露了出来。这使得传统配置数据管理方式在配置数据规模大大增长后基本不可用。

[0084] 另外,在支持简单网络管理协议(以下简称SNMP)的实时通信系统中,除了需要检索、添加、修改、删除配置数据条目以外,还需要各功能模块提供“检索下一个”配置数据条目的功能。在配置数据规模大大增长时,这同样使各功能模块都遇到了时间效率问题。

[0085] 为了解决上述问题,本发明的实施例提供了一种数据管理方法,本发明实施例中使用的RAM数据库的逻辑结构如图1所示,由多个RAM数据表101和各RAM数据表的至少一个索引102组成,每个数据表对应一个嵌入式系统中的功能模块。每个数据表中以记录列表的为单位存储配置数据,记录列表为一数组,结构如图2所示,数组下标即为记录列表的记录编号。图1中的每一个索引都对应图2中的一个索引字段,每个索引又对应一个正向索引表1021和一个反向索引表1022,从图1中可以看出,正向索引表按照该组索引的顺序存储该RAM数据表中各记录列表的记录编号,例如,数据表中存储着记录列表1、2、3、4,其中,按照索引字段T1从小到大的顺序进行排序后,就得到索引T1的正向索引表,该表反映了各记录列表在字段T1上的排序,依次为记录列表2、4、1、3;反向索引表存储该RAM数据表中各记录列表的按该组索引的顺序排序后的记录名次,在正向索引表完成排序后,可知记录列表2的记录名次(即排序后的位置)为1,就在反向索引表的第二个位置上存储1,依此类推。反向索引表仅存储记录列表的记录名次,在系统最大表记录数不超过64K时,反向索引表的每项使用16位即可,可以节省50%的RAM开销。反向索引表在添加、修改、删除记录列表时相应修改,以维护反向索引表的正确性。

[0086] 下面结合上述附图,对使用本发明的实施例一进行说明。本发明的实施例一提供了一种数据管理方法,结合图1及图2所示的RAM数据库,实现了数据存储,具体流程如图3所示,包括:

[0087] 步骤301、分别为嵌入式系统中各功能模块建立对应的RAM数据表;

[0088] 本步骤中,在RAM数据库初始化时,分别确定各RAM数据表的容量、表记录长度、主键和至少一个索引,为所述RAM数据表与至少一个索引分配内存资源。

[0089] 每个RAM数据表必须定义一个主键,用于唯一区分表中的各条记录列表,主键可以包含记录列表的一个或多个字段;每个数据表还可以定义一组或多组索引,用于快速从表中检索。每组索引占用表的一个字段。通常情况下,主键的构成字段都定义为索引。

[0090] 每个数据表的每个索引,对应两个索引表:一个是按各记录的该索引值从小到大的进行排序的记录编号表(以下简称正向索引表);一个是反向的各记录的排序名次表(以下简称反向索引表)。这两个索引表在添加、修改和删除记录列表时更新,在检索时使用。

[0091] 其中,在RAM数据库不支持检索下一个的功能时,可以不创建反向索引表。

[0092] 步骤302、在各个功能模块对应的RAM数据表数据组存储该功能模块的配置数据;

[0093] 本步骤中具体如图4所示,当嵌入式系统的一个功能模块需要存储配置数据时,首先生成一个或多个包含上述配置数据的记录列表,在定义好该记录列表的各个字段后,

将这待存储的记录列表输入到 RAM 数据库中,由 RAM 数据库使用快速排序算法对输入的待存储记录列表进行排序。然后,判断排序后的多个待存储记录列表中是否有两个相邻记录列表中的主键字段完全相同,如果存在,则返回失败消息,在该失败消息中指示待存储的记录列表中有重复的现象存在,还可以携带重复的记录列表的主键或记录编号;否则,进一步的判断在 RAM 数据库中是否存在与待存储的记录列表主键字段完全相同的记录列表,并在存在与该待存储的记录列表主键字段完全相同的记录时,返回失败消息,不存储该记录列表;在不存在主键字段完全相同的记录时,将待加入的记录复制到数据表尾部,其记录编号在原有最大记录编号后依次递增。之后,初始化所有正向索引表,具体为按照记录编号的顺序排序;最后使用快速排序算法对本 RAM 数据表的所有索引字段对应的正向索引表重新排序,并根据重新排序后的正向索引表更新反向索引表。

[0094] 当待存储的记录列表数据较少(如少于 100 条)时,也可以使用插入排序法结合二法,将待存储的记录列表逐条加入到正向索引表中,再更新反向索引表。

[0095] 下面结合附图,对本发明的实施例二进行说明。

[0096] 本发明的实施例二提供了一种数据管理方法,能够对 RAM 数据库中的记录列表进行检索,具体流程如图 5 所示,包括:

[0097] 步骤 501、接收检索记录指令,该指令中包含至少一个检索条件;

[0098] 本步骤中,多个检索条件逻辑相与,一个检索条件具体可以为某字段在指定取值范围之内,或者某字段等于指定值。检索条件中涉及的表字段都属于同一个 RAM 数据表,且已在该字段上定义了索引,检索结果固定为满足检索条件的所有表记录。

[0099] 需要说明的是,实际检索时可以根据需要指定其中的一个或多个索引(索引条件之间是逻辑与的关系),甚至可以不指定任何索引(此时检索结果为所有记录)。

[0100] 步骤 502、根据各检索条件分别进行检索,获取各检索条件的检索结果集合,所述检索条件具体为一个字段的取值范围或指定值;

[0101] 本步骤中,通过二分法分别获取各个检索条件的检索结果集合,其中,需要说明的是,在数据表为空或存在一个无效检索条件(即根据该检索条件检索不到记录列表)时,认为检索失败,返回空集;在 N = 0,即未输入检索条件时,返回 RAM 数据表中记录列表的全集。

[0102] 具体如图 6 所示,输入数据表 T 上的 N 个索引 $\{TI_i \mid i = 1, 2, \dots, N\}$ 以及在这 N 个索引上的取值范围 $\{(T_{min_i}, T_{max_i}) \mid T_{min_i} \text{ 为下界(含)}, T_{max_i} \text{ 为上界(含)}, i = 1, 2, \dots, N\}$;然后判断数据表是否为空,如果为空,则返回空集,如果不为空,则判断 N 是否等于 0;如果 N 为 0,返回全集,否则,r 为指定索引列表中检索结果集最小的索引编号,用于使用二分法完成检测;将 r 和 i 的初始值均设置为 1,然后在正向索引表 TI_i 上用二分法查找第一个大于等于 T_{min_i} 的记录名次 0_{is} 和最后一个小于等于 T_{max_i} 的记录名次 $0_{ie}, 0_{rs}$ 和 0_{re} 分别表示在第 r 索引上的结果记录起始排序名次和结束排序名次;如果没找到符合条件的记录名次,则返回空集,否则,进一步判断 $0_{ie} - 0_{is} < 0_{rs} - 0_{re}$;具体的,若 $0_{ie} - 0_{is} < 0_{rs} - 0_{re}$,则 $r := i$,并将 $i := i + 1$;之后,对 i 和 N 进行比较,具体的,如果 $i <= N$,则继续使用二分法查找,否则统计出各索引上结果集 $\{[0_{is}, 0_{ie}] \mid i = 1, 2, \dots, N\}$ 中不是全集的个数 M,其中 $[0_{is}, 0_{ie}]$ 表示名次在 0_{is} 和 0_{ie} 之间的所有记录;进一步判断 M 的值,若 $M <= 1$,则返回索引 I_r 上的结果集 $[0_{rs}, 0_{re}]$,否则进入步骤 503。

[0103] 步骤 503、取各检索结果集合的交集作为最终检索结果；

[0104] 本步骤中，每个检索条件检索获取的检索结果集合均对应一个位图，位图的第 k bit 对应 RAM 数据表中记录编号为 k 的记录列表，如果该记录列表满足检索条件，则将该 bit 置为 1，否则置为 0。

[0105] 按照上述方法，依次对各个检索结果集合进行处理，将后一检索结果集合累加到其之前的检索结果集合交集中，直至完成全部检索结果集合的处理，得到最终检索结果。

[0106] 具体如图 7 所示，例如，输入数据表 T 在 N 个索引 $\{TI_i | i = 1, 2, \dots, N\}$ 上的结果集区间 $\{[0_{is}, 0_{ie}] | i = 1, 2, \dots, N\}$ ，统计出结果记录个数最小的索引 TI_r ，然后判断 TI_r 上的结果记录个数为 0，如果是则返回空集，否则定义一个位图 $bitmap[0..1]$ ，其中 0、1 为位图编号；位图的第 k bit 表示记录编号为 k 的记录是否在结果集中，将 TI_r 上结果集全部标识到位图 0 上；设置变量 b，用于获取最终检索结果，初始时设置 $b := 0, i := 0$ ，然后根据 $i := i+1$ 对 i 进行处理；将 i 与 N 进行比较，如果 $i > N$ ，则返回 TI_r 结果集中那些在 $bitmap[b]$ 上已标记的记录，否则判断 $i = r$ 或者 TI_i 上的结果集为全集；在判断结果为全集时，对 i 进行 $i := i+1$ 的处理，在判断结果不为全集时，将位图 $bitmap[1-b]$ 清 0，对 $[0_{is}, 0_{ie}]$ 的每个记录，若其在位图 $bitmap[b]$ 中已标记，则在 $bitmap[1-b]$ 中上标记该记录，并根据 $b := 1-b$ 对 b 的值进行处理，再对 i 进行 $i := i+1$ 的处理，并再将 i 与 N 进行比较，以确定是否返回 TI_r 结果集中那些在 $bitmap[b]$ 上已标记的记录。

[0107] 使用位图可以提高检索的总体效率。计算两个分别有 N 和 M 个元素的集合的交集时，通常情况下的效率为 $N*M$ ，在使用位图时，效率为 $N+M$ 。

[0108] 需要说明的是，在同一个 RAM 数据表中，可能存在一个或多个索引值均相同的多个记录列表，故最终检索结果可以是一个记录列表，也可以是多个均满足全部检索条件的记录列表。

[0109] 在步骤 501 至 503 完成检索的基础上，还可以实现对已存储在 RAM 数据表中的记录列表的修改和删除，在检索到需要操作的列表记录后，对其进行修改或删除的步骤如下。

[0110] 步骤 504、修改 RAM 数据表中存储的配置数据；

[0111] 本步骤中，接收到修改指令后，根据修改指令中携带的检索条件检索到相应的记录列表，进行修改替换。如果 RAM 数据表中存在相邻的多条主键字段完全相同的记录列表，则认为 RAM 数据表发生错误，返回失败消息；或在 RAM 数据表中，不存在符合检索条件的记录列表时，返回失败消息。

[0112] 在修改完记录列表后，根据修改后的结果，更新该 RAM 数据表的全部正向索引表及反向索引表。

[0113] 具体如图 8 所示，例如，输入一组要修改的数据表 T 的记录列表，使用快速排序算法对输入的记录列表进行排序，并判断排序后的记录列表中是否存在主键字段完全相同的相邻记录，如果存在则返回失败，如果不存在则调用图 6 所示过程确定待修改的所有记录在数据表中是否已存在；调用图 6 所示过程后，如果判断结果为不存在，则返回失败，否则用待修改的记录替换到数据表中的具有相同主键的记录，然后使用快速排序算法对此表所有非主键成员字段上的正向索引表重新排序（按照所代表的表记录的对应索引值）；根据变化后的所有正向索引表直接更新反向索引表（已知正向索引表中名次为 k 的记录的编号为 D_k ，则反向索引表中编号为 D_k 的记录的名次为 k），返回成功。

[0114] 步骤 505、删除 RAM 数据表中存储的配置数据；

[0115] 本步骤中，接收删除指令，根据删除指令中的检索条件检索相应的记录列表，并在检索到需要删除的记录列表后，执行删除，在删除完成后，还需要对该 RAM 数据表的全部正向索引表及反向索引表进行更新。

[0116] 具体过程如图 9 所示，例如，首先输入数据表 T 上的 N 个索引 $\{TI_i | i = 1, 2, \dots, N\}$ 以及在这 N 个索引上的取值范围 $\{(T_{min_i}, T_{max_i}) | T_{min_i} \text{ 为下界（含）}, T_{max_i} \text{ 为上界（含）}, i = 1, 2, \dots, N\}$ ，然后调用图 6 所示的过程得到满足条件的记录集 R；将表中最后一条记录移动到待删除的 j 记录的存储位置，并相应修改两个索引表中 j 记录的名次 $O_j = O_{last}, D_{O_{last}} = j$ ，其中，表示反向索引表中编号为 j 的记录的名次， O_{last} 表示反向索引表中编号最大的记录的名次， $D_{O_{last}}$ 表示正向索引表中名次为 O_{last} 的记录的编号；最后将反向索引表中名次大于 O_j 的所有记录的名次减 1，返回成功。

[0117] 下面结合附图，对本发明的实施例三进行介绍。

[0118] 本发明的实施例三提供了一种数据管理方法，使用该方法，可以实现嵌入式系统数据库的“检索下一个记录”功能，具体流程如图 10 所示，包括：

[0119] 步骤 1001、接收检索下一个记录指令，该指令中包含至少一个检索条件；

[0120] 步骤 1002、根据各检索条件依次进行检索，分别获取各有效检索条件的检索结果集合；

[0121] 本步骤中，依次按照各个检索条件获取对应的检索结果集合，检索结果不为空的检索条件为有效检索条件，直到发现一个检索条件的检索结果为空集（即该检索条件无效）时，停止检索。

[0122] 具体如图 11 所示，例如，步骤 1101 输入数据表 T 上的 $N > 0$ 个索引 $\{TI_i | i = 1, 2, \dots, N\}$ 以及在这 N 个索引上的取值 $\{Tv_i | i = 1, 2, \dots, N\}$ ，步骤 1102 将 i 设置为 $i = 1$ ，步骤 1103 在正向索引表 TI_i 上用二分法查找第一个大于等于 Tv_i 的记录名次 O_{is} （找不到时为表记录数 +1）和最后一个小于等于 Tv_i 的记录名次 O_{ie} （找不到时为 0）；然后在步骤 1104 中比较 O_{is} 和 O_{ie} ，如果 $O_{is} \leq O_{ie}$ 则进入步骤 1105，否则进入步骤 1108；步骤 1105 中将 TI_i 上结果集全部标识到位图 0 上；步骤 1106 比较 i 和 N 的关系，如果 $i \leq N$ 则返回步骤 1103，否则进入步骤 1107；步骤 1107 根据 $i := i - 1$ 对 i 进行处理；步骤 1108 比较 O_{is} 和 O_{ie} ，如果 $O_{is} \leq O_{ie}$ 则进入步骤 1110，否则进入步骤 1109。

[0123] 步骤 1109 中，比较 O_{is} 和表记录数，如果 $O_{is} >$ 表记录数则进入步骤 1115，否则进入步骤 1110；步骤 1110 中， $O_{ie} :=$ 表记录数，然后进入步骤 1113。

[0124] 步骤 1111 中，比较 O_{ie} 和表记录数，如果 $O_{ie} =$ 表记录数，则进入步骤 1115，否则进入步骤 1112；步骤 1112 中 $O_{is} := O_{ie} + 1, O_{ie} :=$ 表记录数，然后进入步骤 1113。

[0125] 步骤 1113 调用图 12 的过程计算所有 $[O_{js}, O_{je}], j = 1, 2, \dots, I$ 的交集中索引 TI_i 上取值最小的记录集 R，进入步骤 1114；步骤 1114 中判断 R 是否为空集，如果 R 为空集，则进入步骤 1115，否则调用图 13 的过程计算并返回 R 中索引 $\{TI_j, j = i + 1, \dots, N\}$ 值按字典序最小的记录集。

[0126] 步骤 1115 中 $i := i - 1$ ，进入步骤 1116；步骤 1116 中，如果 $i \geq 1$ ，则返回步骤 1118，否则返回空集。

[0127] 在停止检索后，取已获取的各检索结果集合的交集，具体流程如图 12 所示，例如，

步骤 1201 中输入数据表 T 在 N 个索引 $\{TI_i | i = 1, 2, \dots, N\}$ 上的结果集区间 $\{[0_{is}, 0_{ie}] | i = 1, 2, \dots, N\}$;步骤 1202 统计出结果记录个数最小的索引 TI_r ;步骤 1203 中判断 TI_r 上的结果记录个数是否为 0,如果为 0 则返回空集,否则进入步骤 1204;步骤 1204 中定义一个位图 $bitmap[0..1]$,其中 0、1 为位图编号,位图的第 k bit 表示记录编号为 k 的记录是否在结果集中;步骤 1205 将 TI_r 上结果集全部标识到位图 0 上;步骤 1206 中令 $b := 0, i := 0$;步骤 1207 中令 $i := i + 1$;步骤 1208 中 $i > N$ 的话进入步骤 1213,否则进入步骤 1209;步骤 1209 判断 $i = r$ 或者 TI_i 上的结果集是否为全集,如果是则进入步骤 1211,否则进入步骤 1210;步骤 1210 将位图 $bitmap[1-b]$ 清 0;步骤 1211 对 $[0_{is}, 0_{ie}]$ 的每个记录,若其在位图 $bitmap[b]$ 中已标记,则在 $bitmap[1-b]$ 中标记该记录(前 i 个索引上的交集);步骤 1212 进行 $b := 1-b$ 的处理,然后回到步骤 1207。

[0128] 步骤 1213 中判断总交集 R 是否为空集,如果是返回空集,如果不是则进入步骤 1214;步骤 1214 中从总交集 R 中找到第 N 索引上名次最小的记录,该记录的第 N 索引值设为 Tv_N ;步骤 1215 在正向索引表 TI_N 上用二分法查找第一个大于等于给定记录 Tv_N 的记录名次 0_{bn} 和最后一个小于等于给定记录 Tv_N 的记录名次 0_{in} (都必能找到),返回 R 与区间 $[0_{bn}, 0_{in}]$ 的交集。

[0129] 步骤 1003、查询最后一个有效检索条件对应的一个索引中的反向索引表,获取所述检索结果集合中的各记录列表的记录名次,并将记录名次最小的记录列表作为检索下一个记录的检索结果;

[0130] 本步骤中,首先获取最后一个有效检索条件对应的反向索引表,将该反向索引表中存储的记录名次最小的记录列表作为检索下一个记录的检索结果。

[0131] 具体如图 13 所示,例如,步骤 1301 输入数据表 T 的记录列表 $R = \{i | i \text{ 为记录编号}, i = 1, \dots, M\}$ 以及 N 个索引 $\{TI_i | i = 1, 2, \dots, N\}$,然后步骤 1302 进行 $i := 1$ 的处理;步骤 1303 利用 TI_i 的反向索引表得到 R 中各记录的名次列表 $L = \{0_j | j = 1, \dots, M\}$;步骤 1304 统计出 L 中的最小名次 0_k ,设该记录的第 i 索引值设为 Tv_i ;步骤 1305 在正向索引表 TI_i 上用二分法查找第一个大于等于 Tv_i 的记录名次 0_{is} 和最后一个小于等于 Tv_i 的记录名次 0_{ie} (都必能找到);步骤 1306 进行 $R := R$ 与区间 $[0_{is}, 0_{ie}]$ 的交集的处理;步骤 1307 进行 $i := i + 1$ 的处理;步骤 1308 比较 i 和 N,如果 $i > N$,则返回 R,否则进入步骤 1303。

[0132] 特别的,当记录名次最小的记录列表为两个或两个以上时,可以在这多个记录列表中,根据前一有效检索条件的反向索引表,获取前一有效检索条件记录名次最小的记录列表作为检索结果,依次类推,直到检索结果仅为一个记录列表,或已使用全部的有效检索条件进行筛选为止。

[0133] 下面结合附图,对本发明的实施例四进行说明。

[0134] 本发明的实施例四提供了一种数据管理方法,建立一个 RAM 数据库和一个 NVRAM 数据库,通过将这数据在这两个数据库中进行同步备份,提高了数据存储的可靠性,具体流程如图 14 所示,包括:

[0135] 步骤 1401、建立 NVRAM 数据库,在所述 NVRAM 数据库中分别为嵌入式系统中各功能模块建立对应的 NVRAM 数据表;

[0136] 本发明的实施例使用的 NVRAM 数据库结构如图 15 所示,NVRAM 数据库中仅存储数据表的当前记录和容量,不存储主键、索引定义以及索引表

[0137] NVRAM 数据表的创建在系统初始化时进行,数据表结构系统运行期间保持不变。RAM 数据库系统初始化后,在 NVRAM 数据库系统初始化时,若不存在某个 NVRAM 数据表,则根据 RAM 数据库的该表信息创建此 NVRAM 表,为该表分配 NVRAM 空间。

[0138] 步骤 1402、在所述 NVRAM 数据库与嵌入式系统中各功能模块对应的 RAM 数据表之间进行记录列表同步;

[0139] 本步骤中,首先将 RAM 数据表记录同步到对应 NVRAM 数据表,为避免向 NVRAM 烧写没有变化的数据表, RAM 数据库为每个 RAM 数据表提供一个同步标志。在对某一 RAM 数据表中的记录列表进行添加、修改、删除时,将该 RAM 数据表的同步标志置为未同步状态。在进行 RAM 数据表到 NVRAM 数据表的同步时,只对同步标志为未同步状态的 RAM 数据表进行同步操作,同步标志为已同步时直接返回操作成功。

[0140] 在将 RAM 数据表记录同步到对应 NVRAM 数据表之后,再将 NVRAM 数据表记录同步到对应 RAM 数据表中,首先采用直接读取裸介质的方式将所有 NVRAM 数据表记录读取到内存中,并删除对应的 RAM 数据表的所有记录,再调用“添加 RAM 数据表记录”过程将上面读取的 NVRAM 数据表加入到对应的 RAM 数据表。

[0141] 在系统初始化时,由于 RAM 数据表空白,故一般在系统初始化时,会执行由 NVRAM 数据表向 RAM 数据表的同步。

[0142] 同步时可以采用直接写裸介质的方式写 NVRAM 数据表,缩短烧写时间。

[0143] 本发明的实施例提供了一种数据管理方法,通过精简关系数据库系统的功能,使用正向和反向索引表,为所有功能模块集中解决了配置数据检索时间效率问题,使 RAM 数据库检索效率达到 $\log(n)$ 的数量级,其中 n 为数据表的实际记录数;对常见的单表容量几万条、索引个数较少的中小型实时通信系统来说,这使得数据库检索时间几乎变为常量。

[0144] 本发明的实施例二和三中,使用位图对多个检索结果集合取交集,实现了取交集的线性效率。而本发明的实施例四中,通过关系数据表格式存储的 NVRAM 数据库,也提高了配置数据存储空间的效率,在这一过程中还使用了直接读写 NVRAM 裸介质的方式,缩短了访问时间,降低了系统在写 NVRAM 期间所占用的系统资源。

[0145] 此外,通过使用正向和反向索引表,也实现了嵌入式系统数据库的使 RAM 数据库“检索下一个”的功能,为所有功能模块解决了 SNMP 中对配置数据的“检索下一个”功能的时间效率问题。

[0146] 本发明的实施例提供的数据管理方法,还以关系数据表格式存储 NVRAM 数据库,极大的提高了配置数据存储空间效率,并采用直接读写 NVRAM 裸介质的方式,缩短了访问时间,降低了系统在写 NVRAM 期间对系统其它功能的影响。使用本发明的实施例提供的数据管理方法对存储及管理配置数据,数据库系统目标代码不超过 30Kbyte,大大减少了硬件资源的占用,能够有效降低设备成本。

[0147] 本发明的实施例还提供了一种数据管理装置,其结构如图 16 所示,包括:

[0148] RAM 数据表建立模块 1601,用于分别为嵌入式系统中各功能模块建立对应的 RAM 数据表;

[0149] 数据存储模块 1602,用于在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据。

[0150] 进一步的,上述数据管理装置如图 17 所示,还包括:

[0151] 检索记录模块 1603，用于接收检索记录指令，该指令中包含至少一个检索条件，根据各检索条件分别进行检索，获取各检索条件的检索结果集合，所述检索条件具体为一个字段的取值范围或指定值，取各检索结果集合的交集作为检索记录的检索结果；

[0152] 检索下一个记录模块 1604，用于接收检索下一个记录指令，该指令中包含至少一个检索条件，根据所述检索条件依次进行检索，分别获取各有效检索条件的检索结果集合，查询最后一个有效检索条件对应的一个索引中的反向索引表，获取所述检索结果集合中的各记录列表的记录名次，并将记录名次最小的记录列表作为检索下一个记录的检索结果；

[0153] 修改模块 1605，用于接收修改指令，所述修改指令中包含至少一个检索条件，根据所述检索条件，检索到相应的记录列表，对所述记录列表进行修改；

[0154] 删除模块 1606，用于接收删除指令，所述删除指令中包含至少一个检索条件，根据所述检索条件，检索到相应的记录列表，对所述记录列表进行修改。

[0155] 本发明的实施例还提供了一种数据管理系统，包括数据管理装置；

[0156] 所述数据管理装置，用于分别为嵌入式系统中各功能模块建立对应的 RAM 数据表，并在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据。

[0157] 进一步的，上述数据管理系统还包括 NVRAM 数据库；

[0158] 所述 NVRAM 数据库，用于分别为嵌入式系统中各功能模块建立对应的 NVRAM 数据表，并在所述 NVRAM 数据库与嵌入式系统中各功能模块对应的 RAM 数据表之间进行记录列表同步。

[0159] 进一步的，所述数据管理装置，还用于接收检索记录指令，该指令中包含至少一个检索条件，根据各检索条件分别进行检索，获取各检索条件的检索结果集合，所述检索条件具体为一个字段的取值范围或指定值，取各检索结果集合的交集作为检索记录的检索结果；和 / 或

[0160] 接收检索下一个记录指令，该指令中包含至少一个检索条件，根据所述检索条件依次进行检索，分别获取各有效检索条件的检索结果集合，查询最后一个有效检索条件对应的一个索引中的反向索引表，获取所述检索结果集合中的各记录列表的记录名次，并将记录名次最小的记录列表作为检索下一个记录的检索结果；和 / 或

[0161] 接收修改指令，所述修改指令中包含至少一个检索条件，根据所述检索条件，检索到相应的记录列表，对所述记录列表进行修改；和 / 或

[0162] 接收删除指令，所述删除指令中包含至少一个检索条件，根据所述检索条件，检索到相应的记录列表，对所述记录列表进行修改。

[0163] 上述数据管理装置和数据管理系统，可以与本发明的实施例提供的一种数据管理方法相结合，在 RAM 数据库中分别为嵌入式系统中各功能模块建立对应的 RAM 数据表，在各个功能模块对应的 RAM 数据表数据组存储该功能模块的配置数据，实现了对嵌入式系统中各个功能块配置数据格式的统一管理，使得数据库结构更简单，简化了实现方式，提高了数据存储效率。此外，还实现了嵌入式系统中的“检索下一个记录”功能，进一步提高了嵌入式系统中配置数据管理的效率。

[0164] 本领域普通技术人员可以理解实现上述实施例方法携带的全部或部分步骤是可以通过程序来指令相关的硬件完成，所述的程序可以存储于一种计算机可读存储介质中，该程序在执行时，包括方法实施例的步骤之一或其组合。

[0165] 另外,在本发明各个实施例中的各功能单元可以采用硬件的形式实现,也可以采用软件功能模块的形式实现。所述集成的模块如果以软件功能模块的形式实现并作为独立的产品销售或使用时,也可以存储在一个计算机可读取存储介质中。

[0166] 上述提到的存储介质可以是只读存储器,磁盘或光盘等。

[0167] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以权利要求所述的保护范围为准。

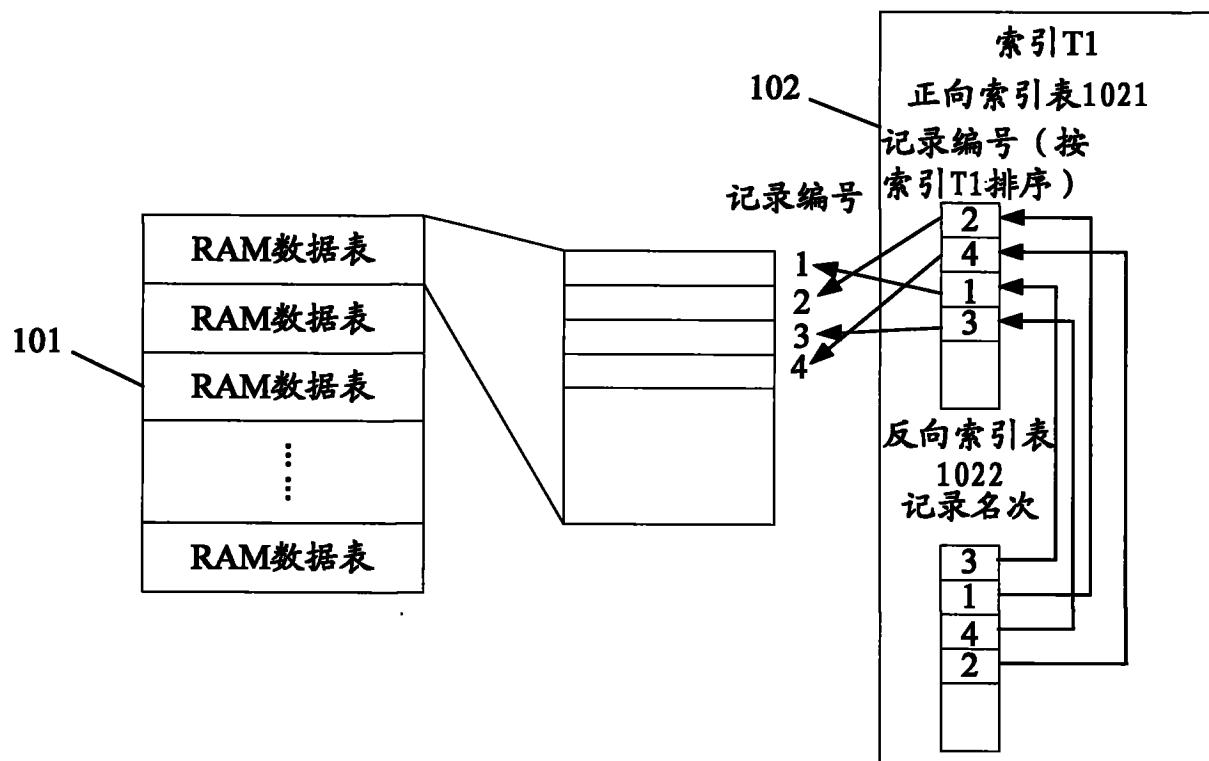


图 1

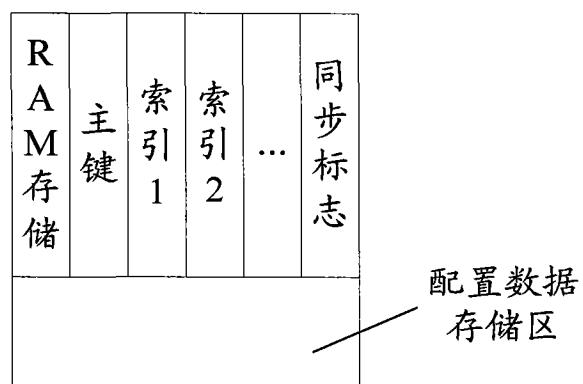


图 2

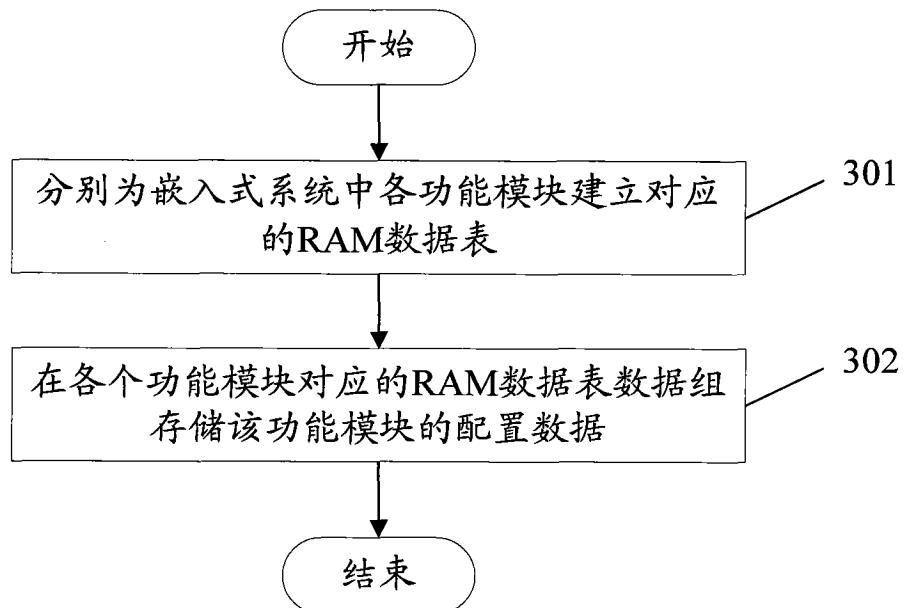


图 3

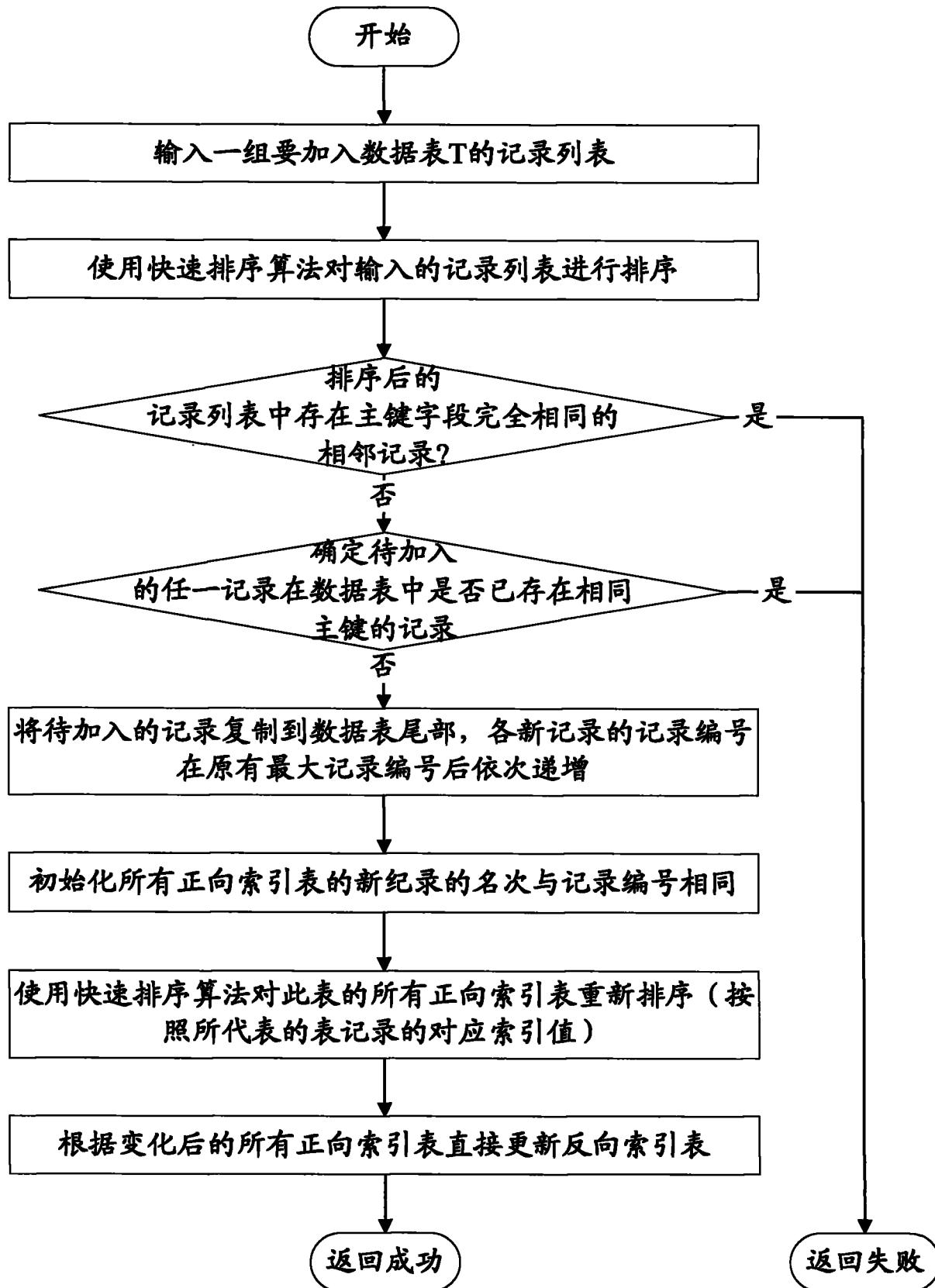


图 4

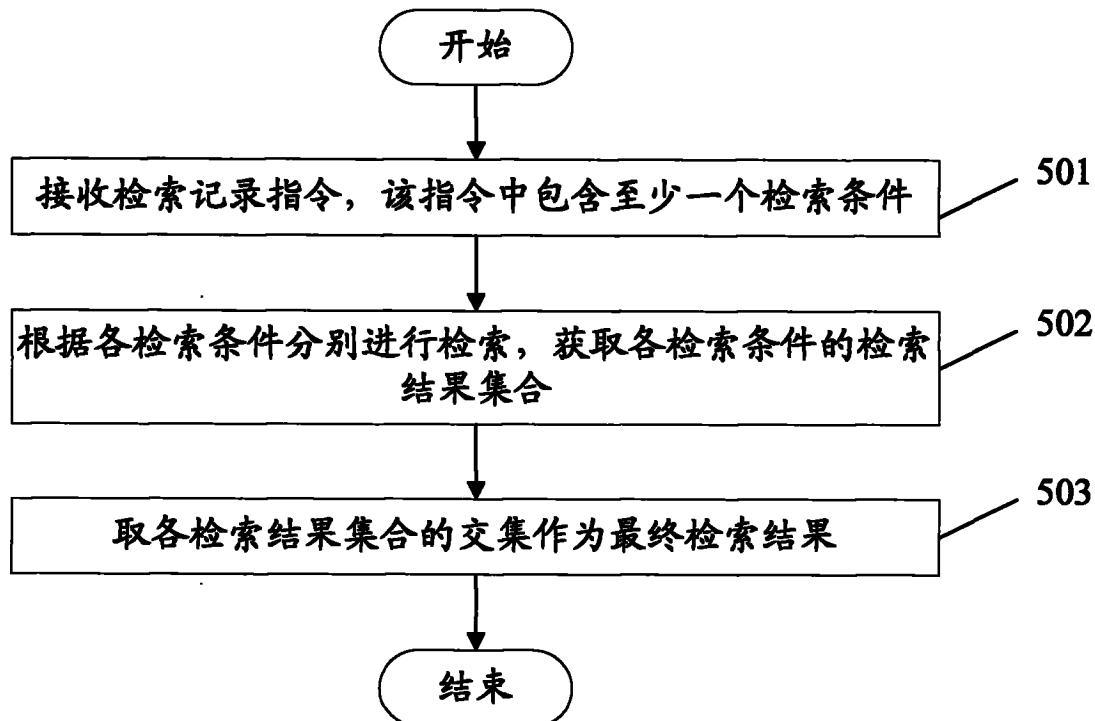


图 5

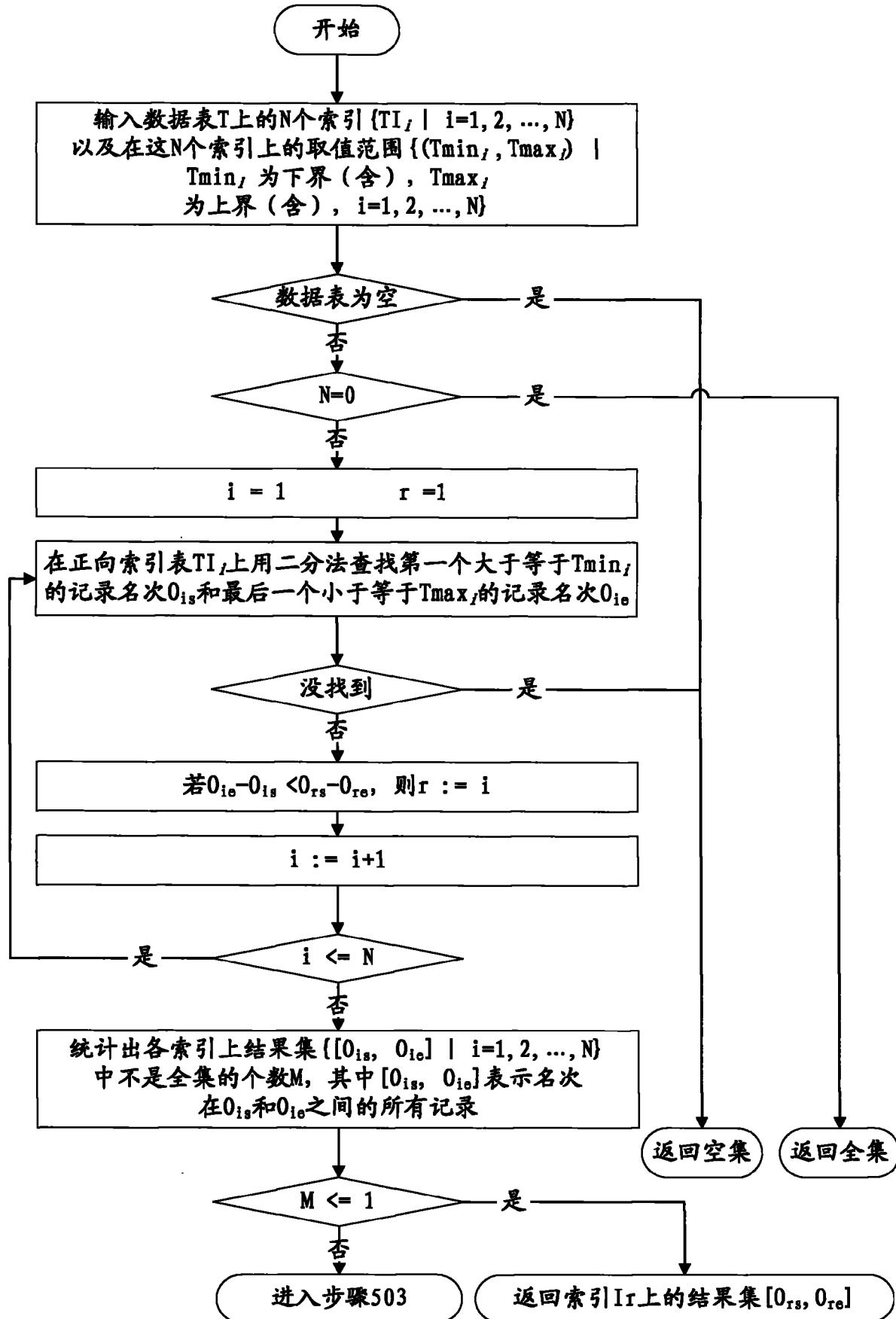


图 6

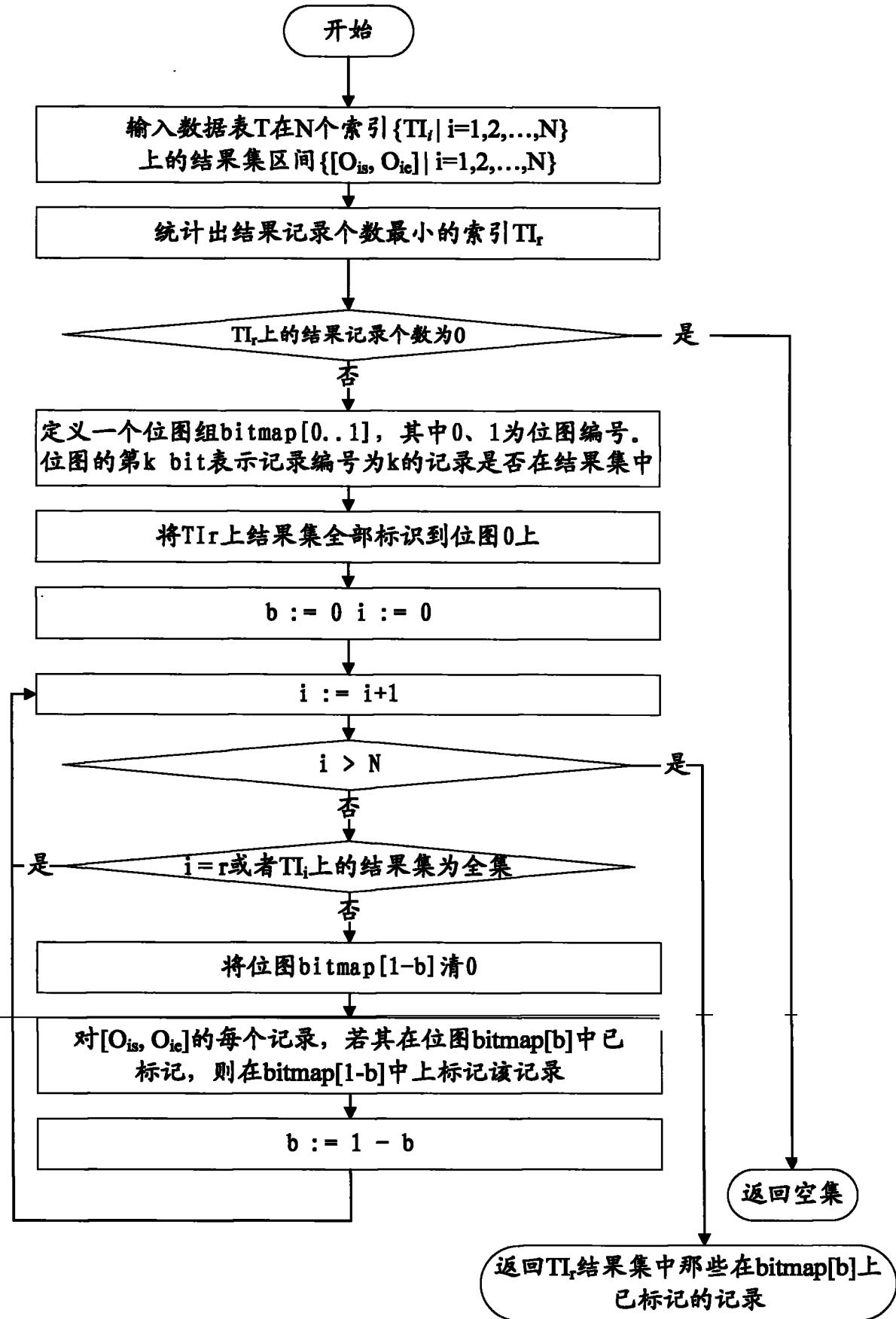


图 7

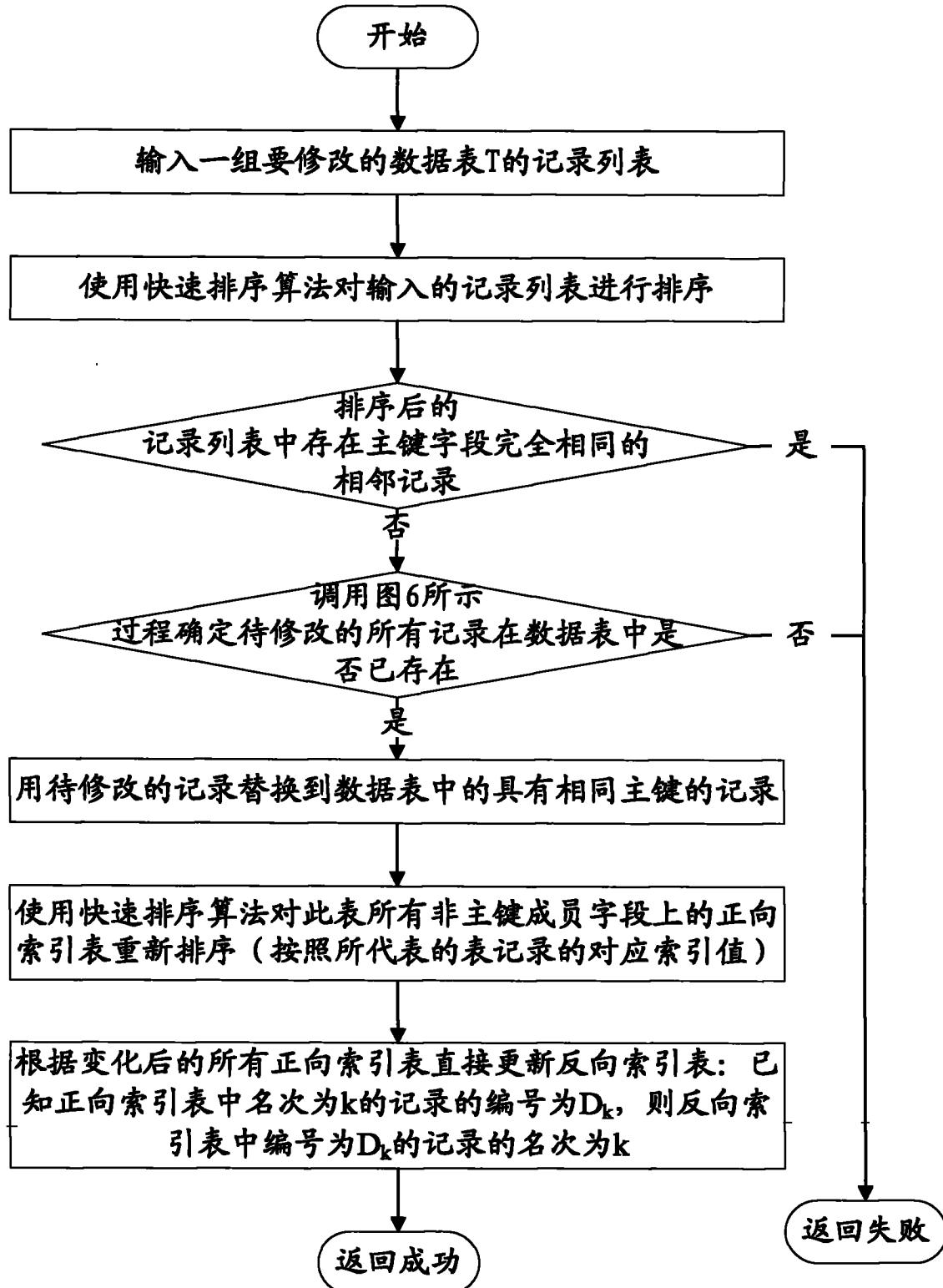


图 8

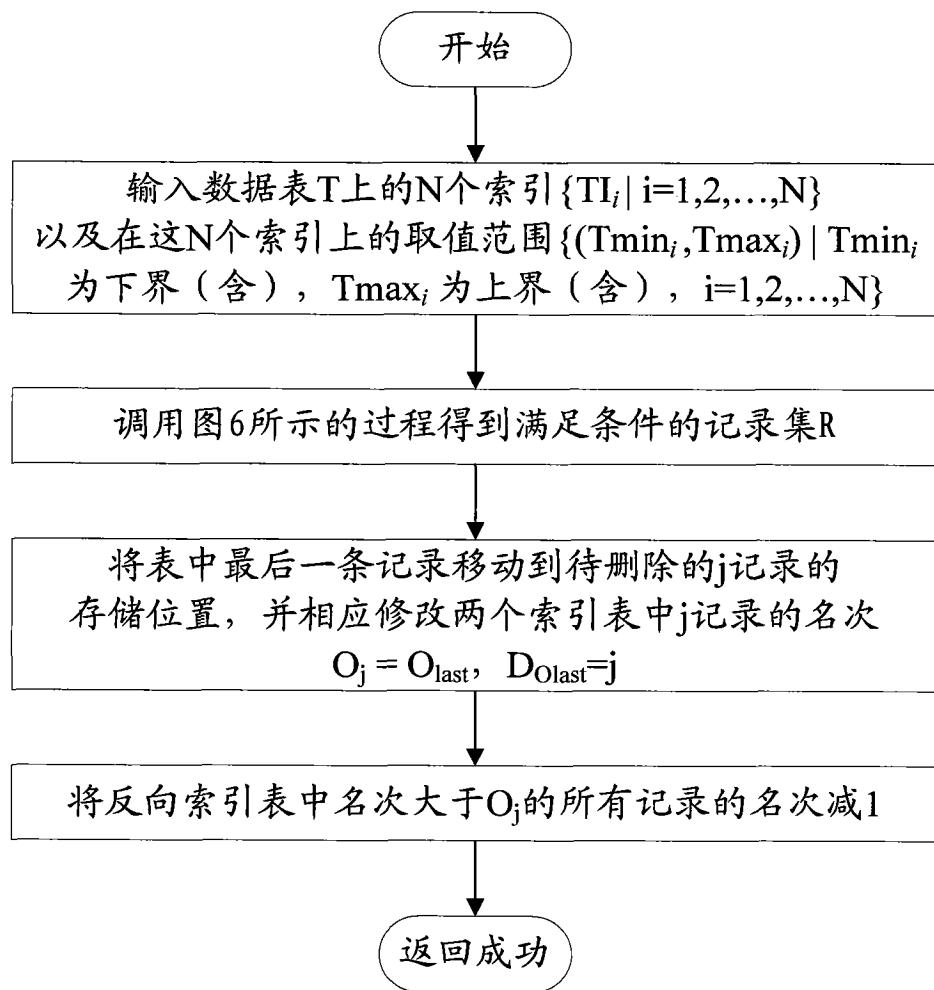


图 9

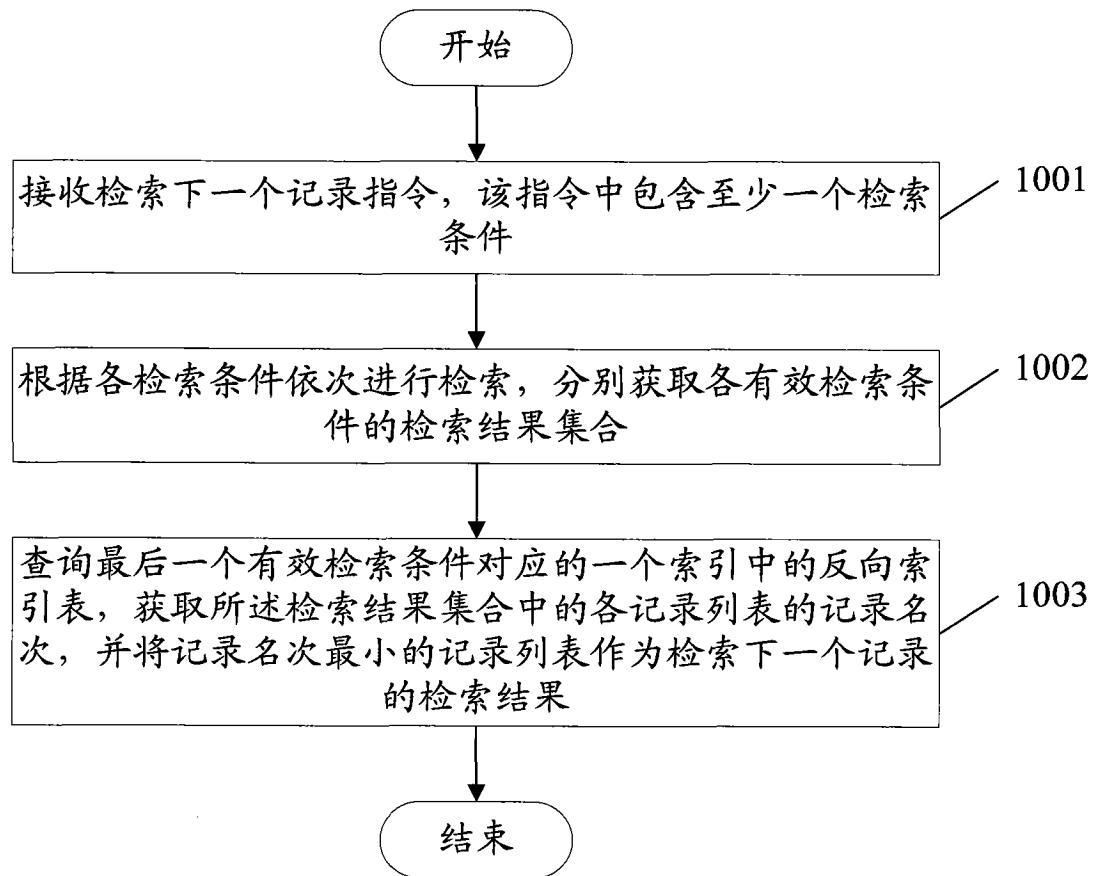


图 10

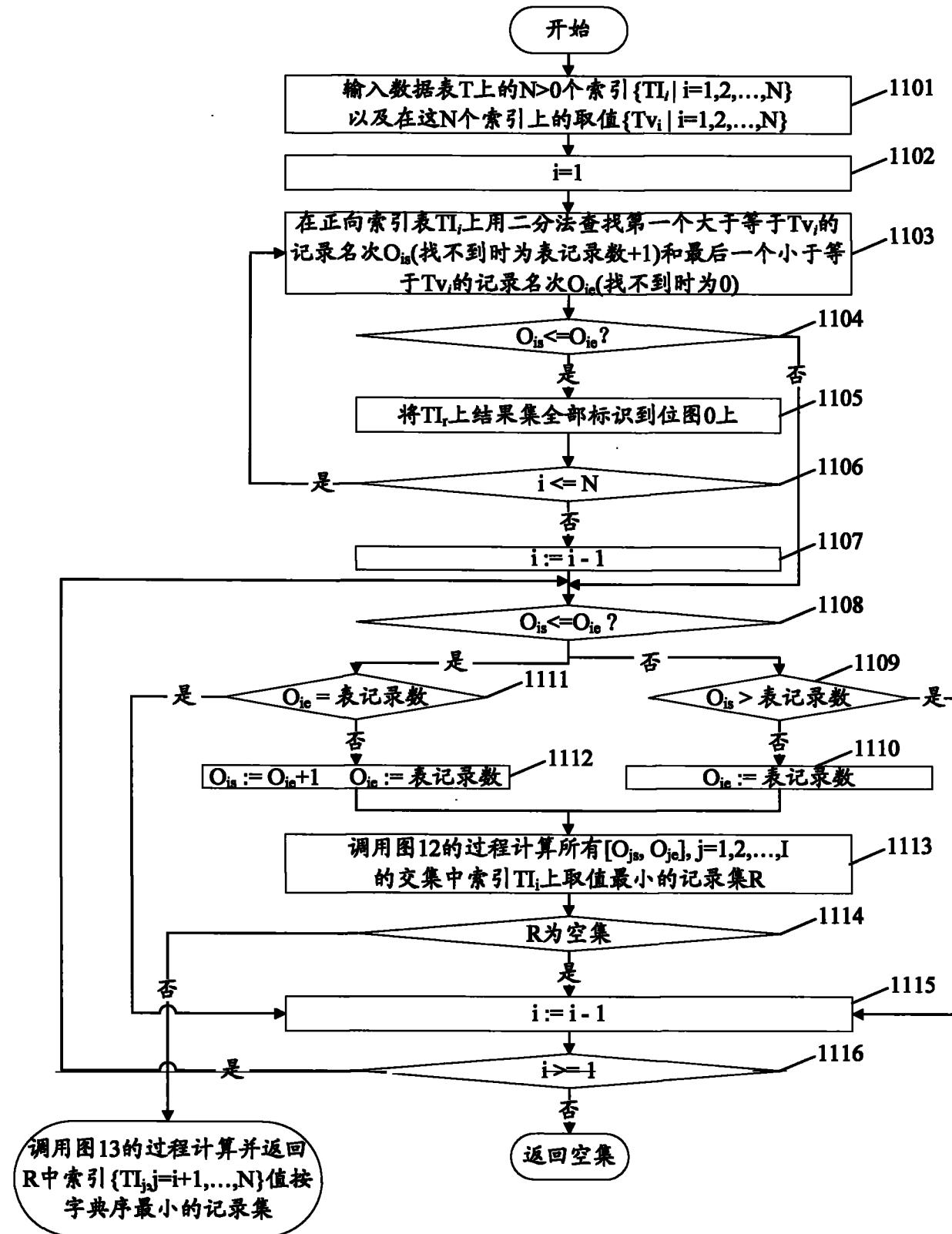


图 11

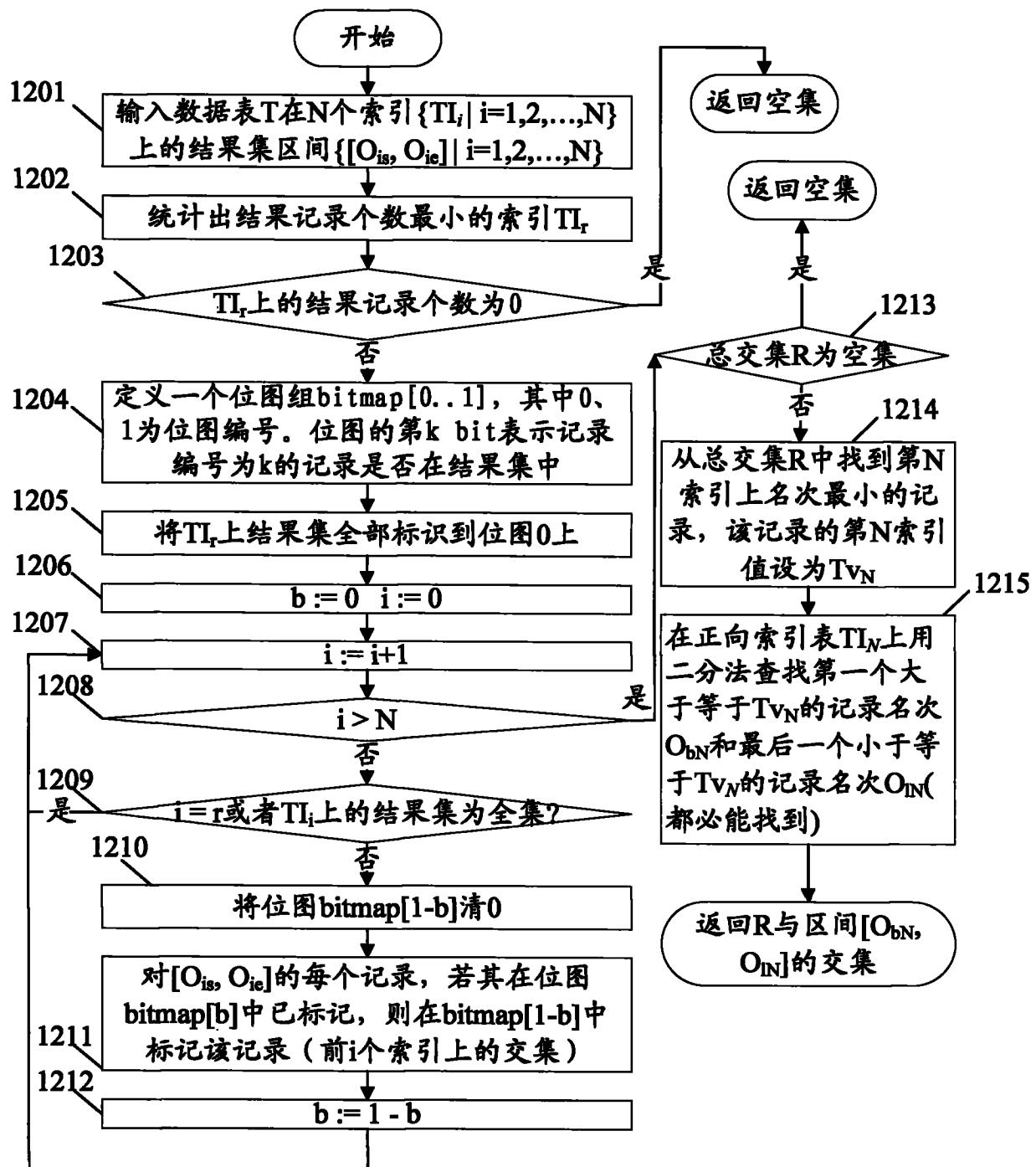


图 12

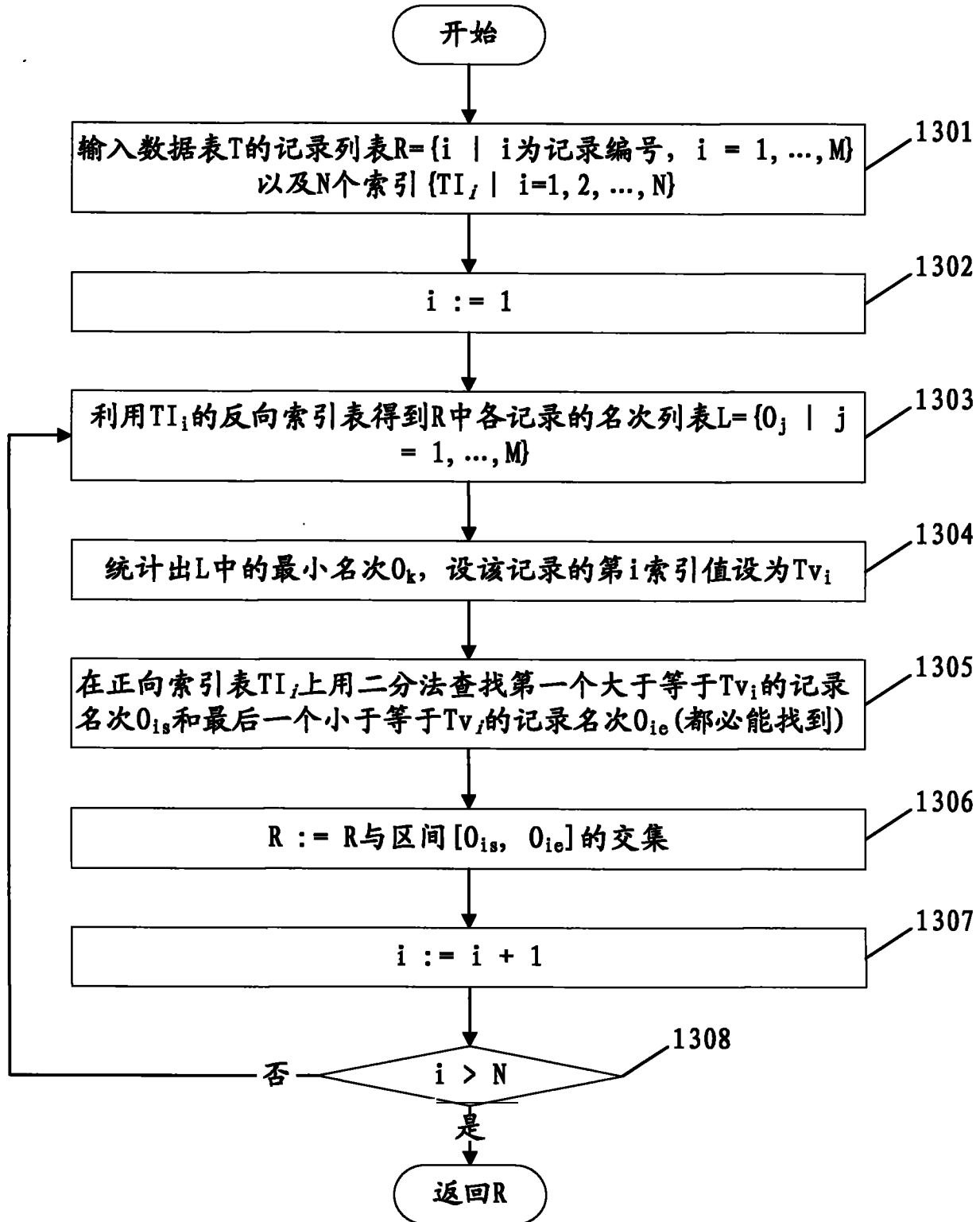


图 13

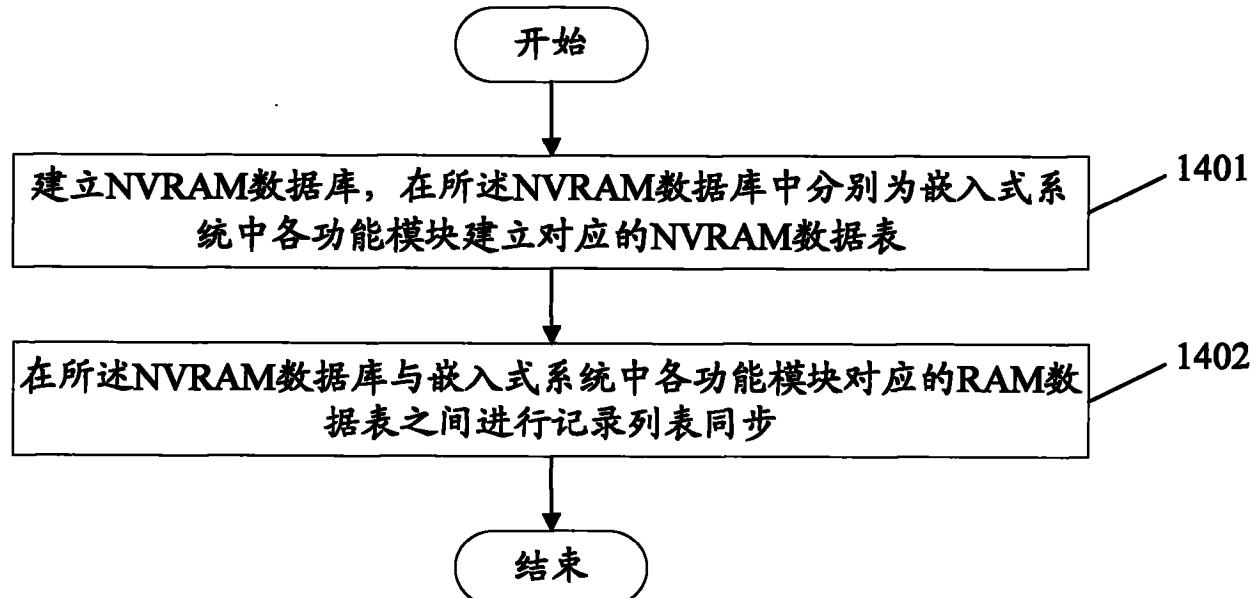


图 14

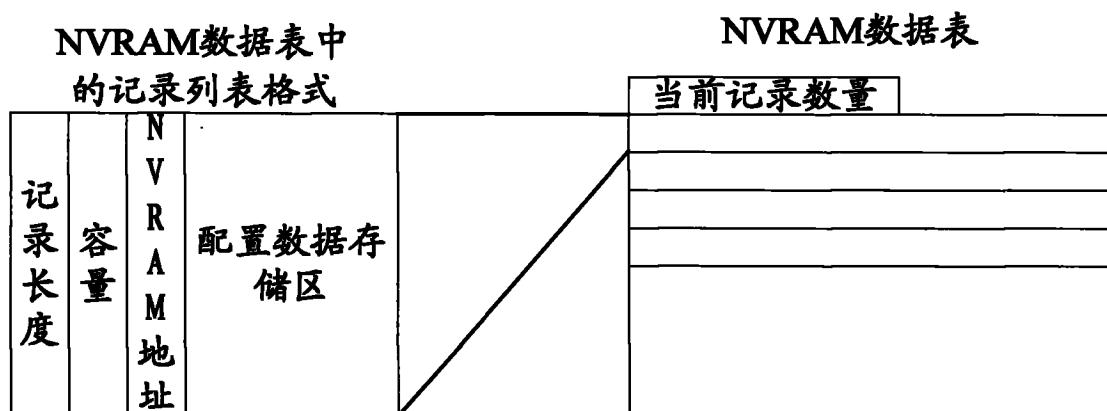


图 15

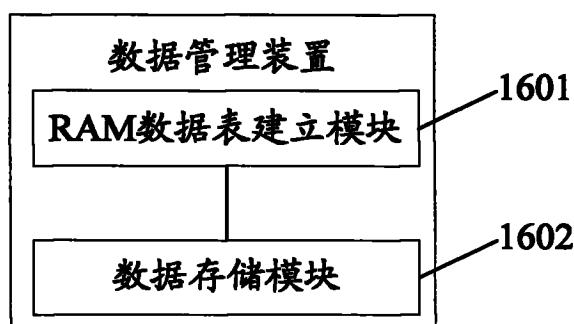


图 16

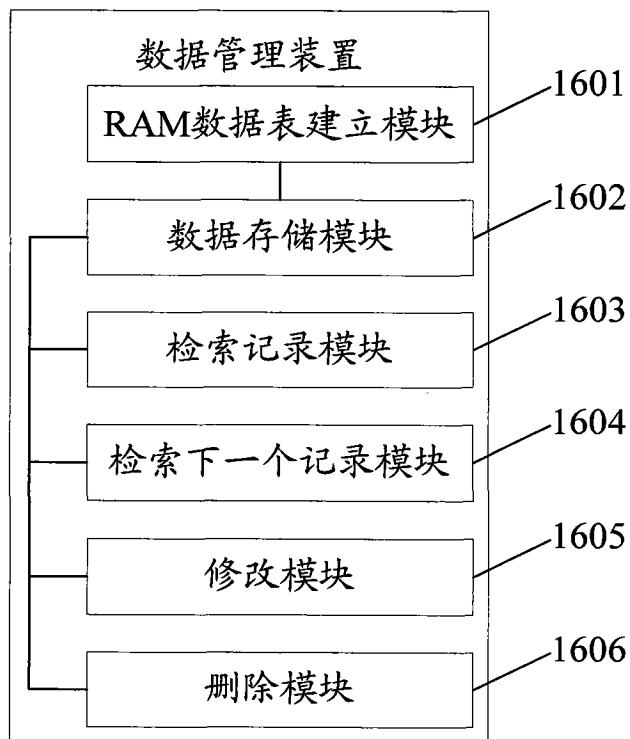


图 17