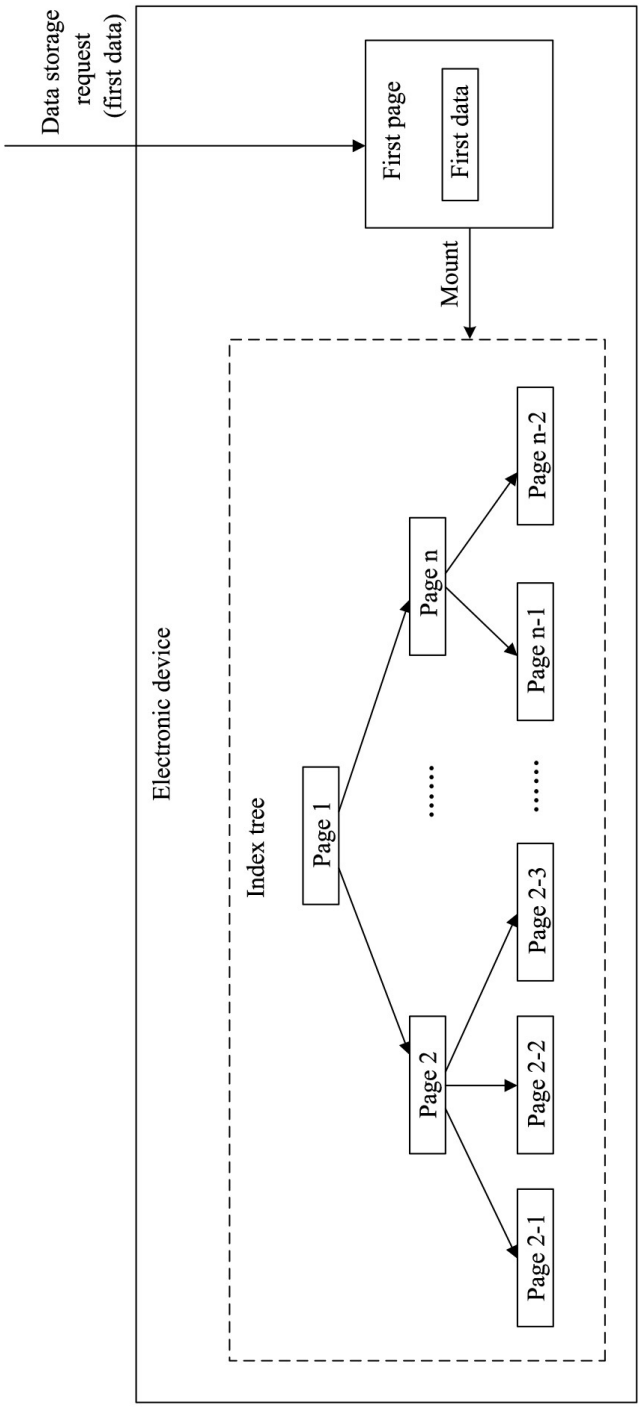

Data storage request (first data)
Electronic device
First page
First data
Mount
Index tree
Page 1
Page 2
Page n
Page 2-1
Page 2-2
Page 2-3
Page n-1
Page n-2
.....
.....

FIG. 1

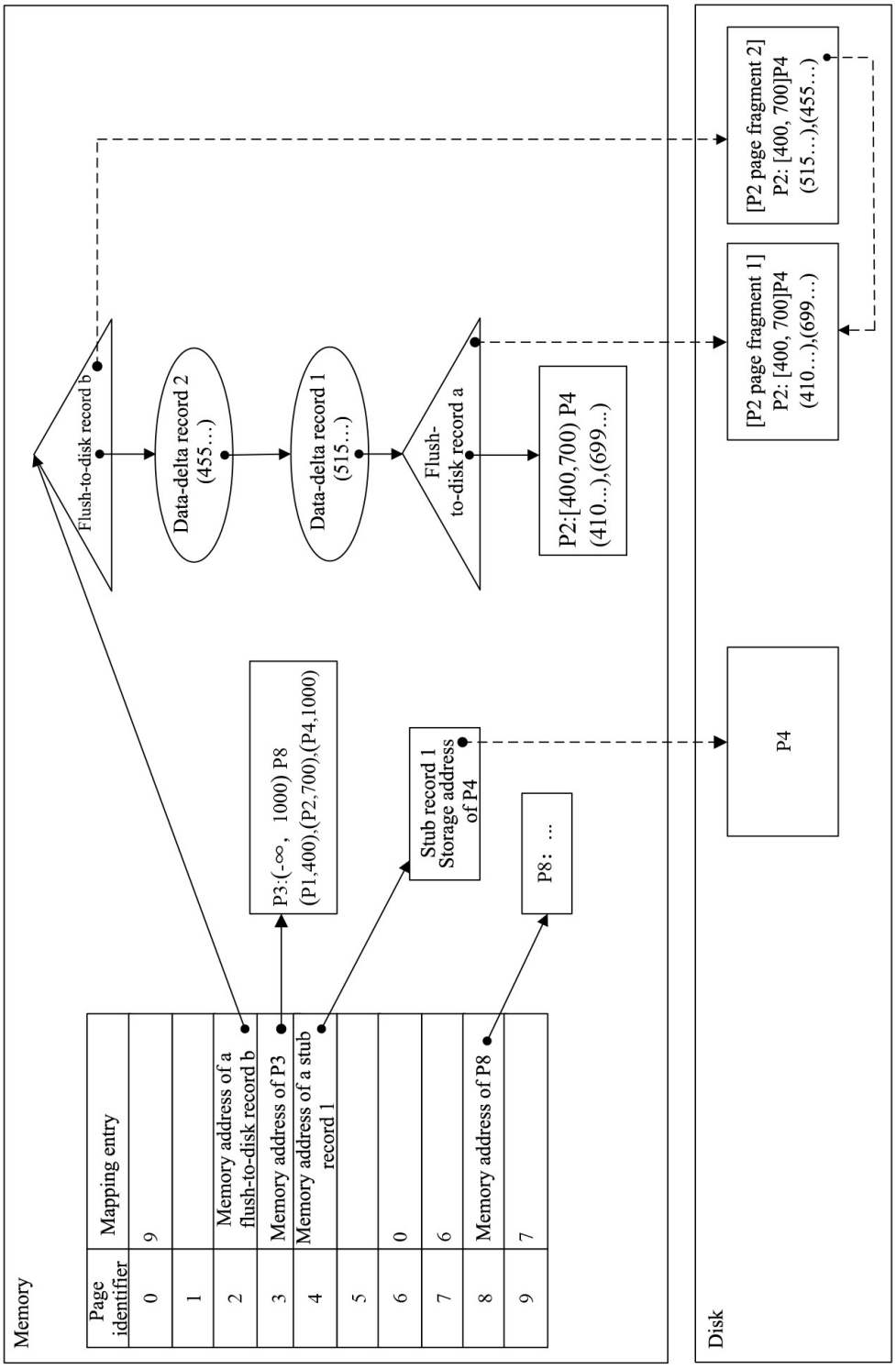

Memory
Page identifier
Mapping entry
0
9
1
2
Memory address of a flush-to-disk record b
3
Memory address of P3
4
Memory address of a stub record 1
5
6
0
7
6
8
Memory address of P8
9
7
Flush-to-disk record b
Data-delta record 2 (455…)
Data-delta record 1 (515…)
Flush-to-disk record a
P2:[400,700) P4 (410...),(699...)
P3:(-∞, 1000) P8 (P1,400),(P2,700),(P4,1000)
Stub record 1 Storage address of P4
P8: ...
Disk
P4
[P2 page fragment 1] P2: [400, 700]P4 (410…),(699…)
[P2 page fragment 2] P2: [400, 700]P4 (515…),(455…)

FIG. 2

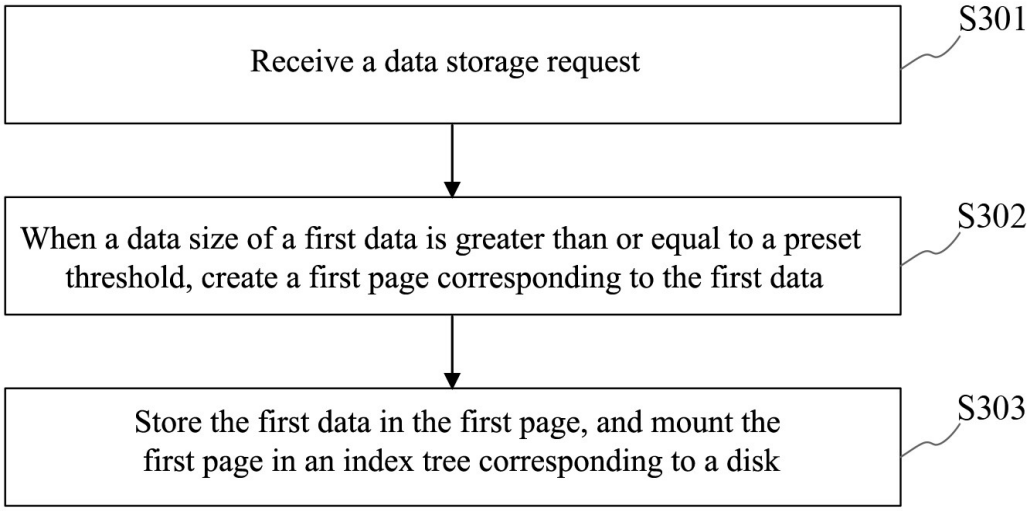

Receive a data storage request
S301
When a data size of a first data is greater than or equal to a preset threshold, create a first page corresponding to the first data
S302
Store the first data in the first page, and mount the first page in an index tree corresponding to a disk
S303

FIG. 3

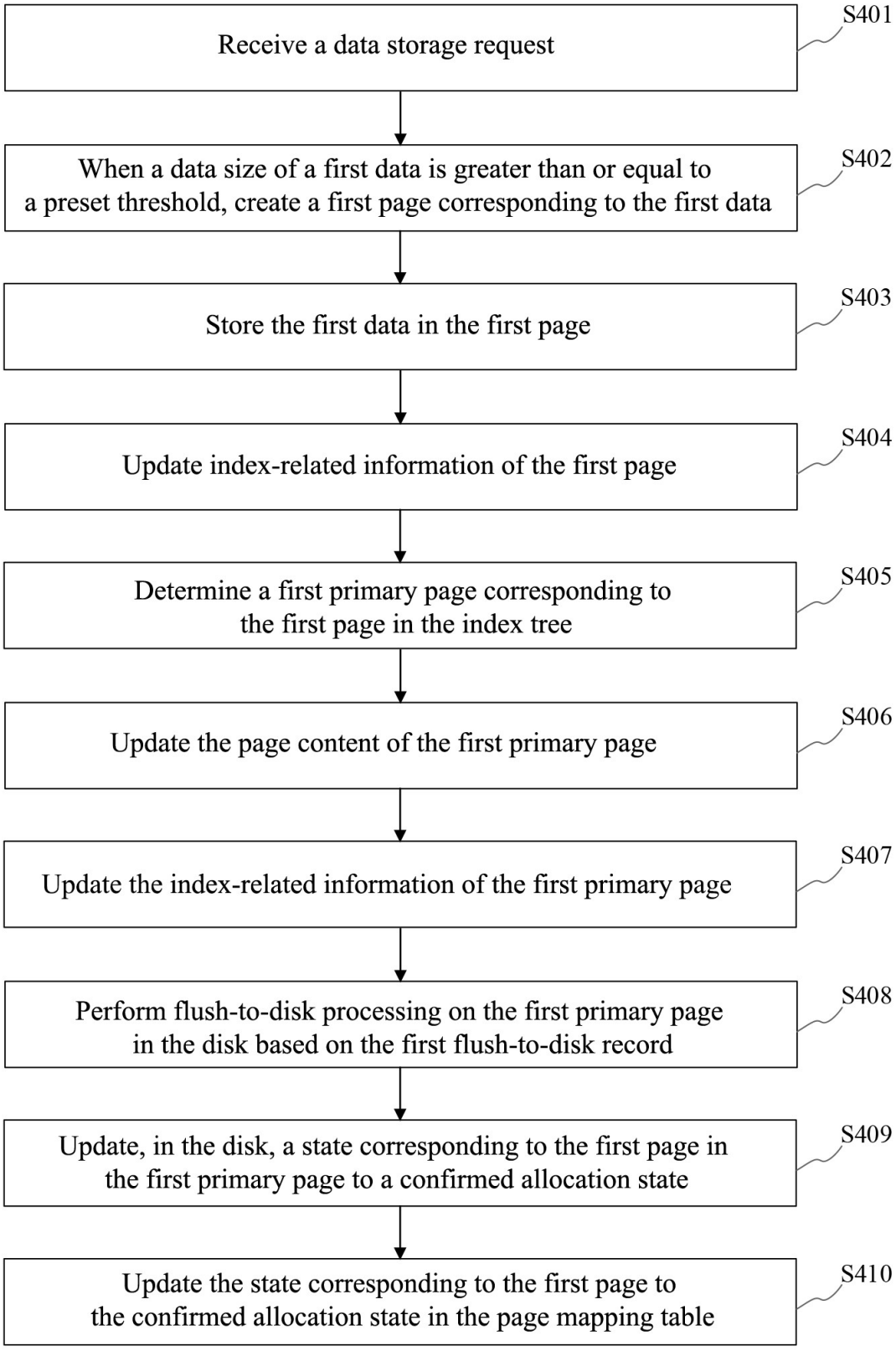

Receive a data storage request
S401
When a data size of a first data is greater than or equal to a preset threshold, create a first page corresponding to the first data
S402
Store the first data in the first page
S403
Update index-related information of the first page
S404
Determine a first primary page corresponding to the first page in the index tree
S405
Update the page content of the first primary page
S406
Update the index-related information of the first primary page
S407
Perform flush-to-disk processing on the first primary page in the disk based on the first flush-to-disk record
S408
Update, in the disk, a state corresponding to the first page in the first primary page to a confirmed allocation state
S409
Update the state corresponding to the first page to the confirmed allocation state in the page mapping table
S410

FIG. 4

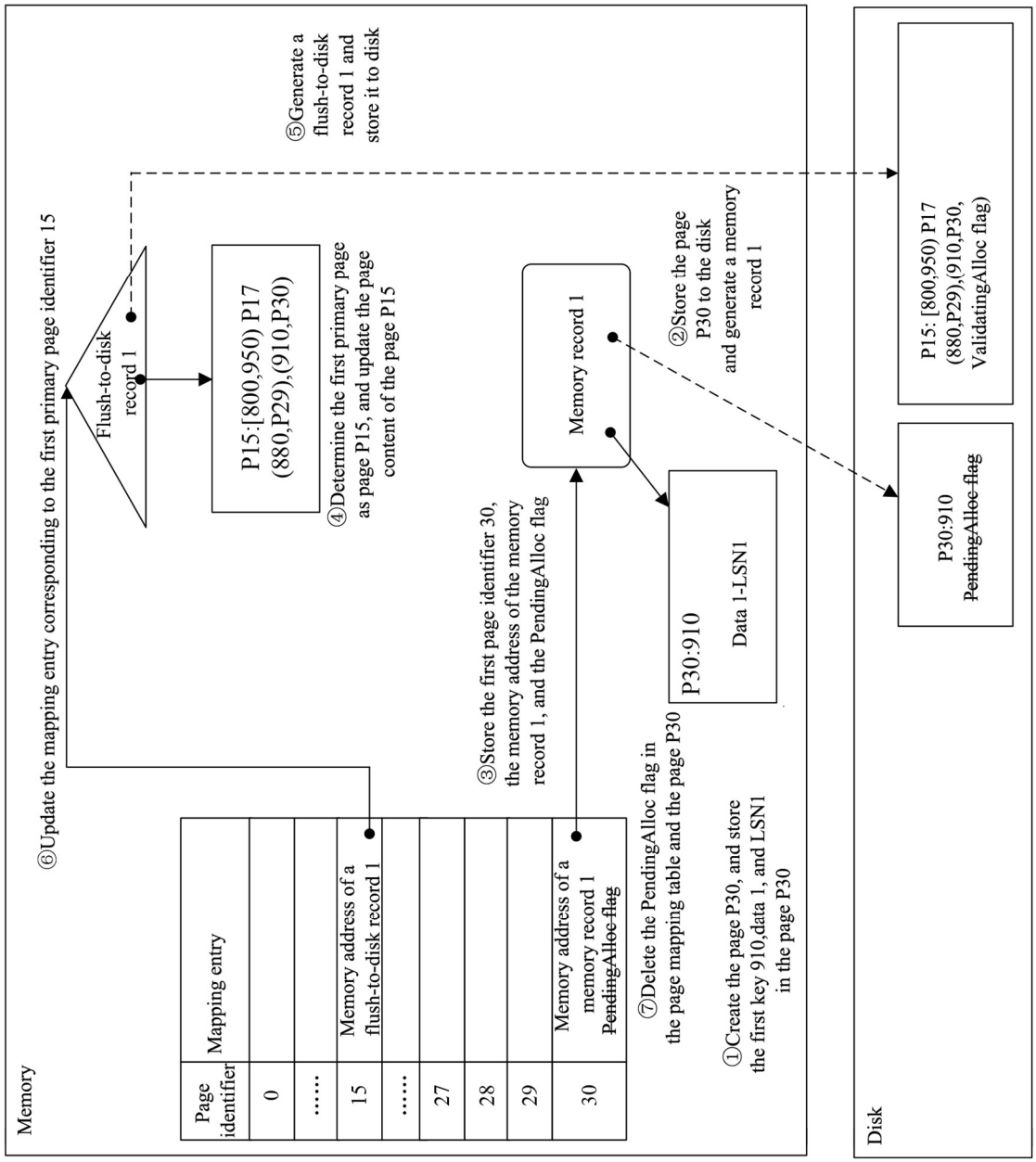

Memory
Page identifier
Mapping entry
0
.....
15
Memory address of a flush-to-disk record 1
.....
27
28
29
30
Memory address of a memory record 1 PendingAlloc flag
⑥Update the mapping entry corresponding to the first primary page identifier 15
Flush-to-disk record 1
P15:[800,950) P17 (880,P29),(910,P30)
④Determine the first primary page as page P15, and update the page content of the page P15
⑤Generate a flush-to-disk record 1 and store it to disk
③Store the first page identifier 30, the memory address of the memory record 1, and the PendingAlloc flag
Memory record 1
②Store the page P30 to the disk and generate a memory record 1
⑦Delete the PendingAlloc flag in the page mapping table and the page P30
P30:910
Data 1-LSN1
①Create the page P30, and store the first key 910,data 1, and LSN1 in the page P30
Disk
P30:910
PendingAlloc flag
P15: [800,950) P17 (880,P29),(910,P30, ValidatingAlloc flag)

FIG. 5

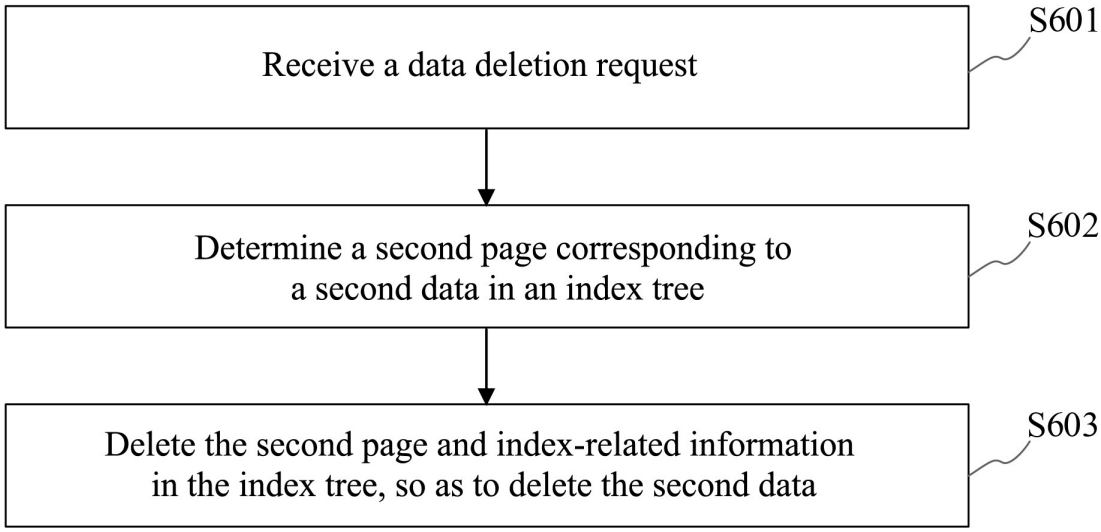

Receive a data deletion request
S601
Determine a second page corresponding to a second data in an index tree
S602
Delete the second page and index-related information in the index tree, so as to delete the second data
S603

FIG. 6

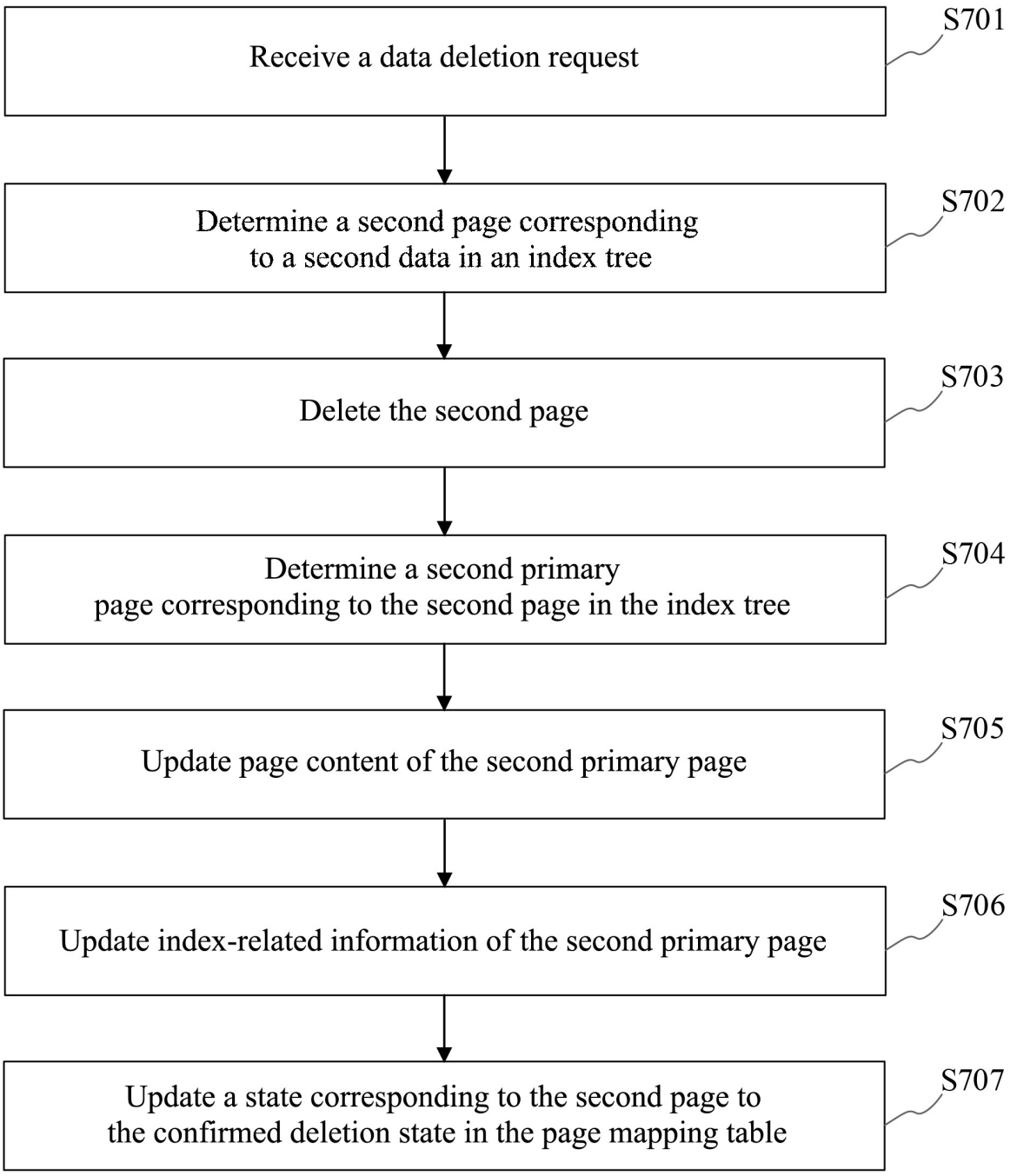

Receive a data deletion request
S701
Determine a second page corresponding to a second data in an index tree
S702
Delete the second page
S703
Determine a second primary page corresponding to the second page in the index tree
S704
Update page content of the second primary page
S705
Update index-related information of the second primary page
S706
Update a state corresponding to the second page to the confirmed deletion state in the page mapping table
S707

FIG. 7

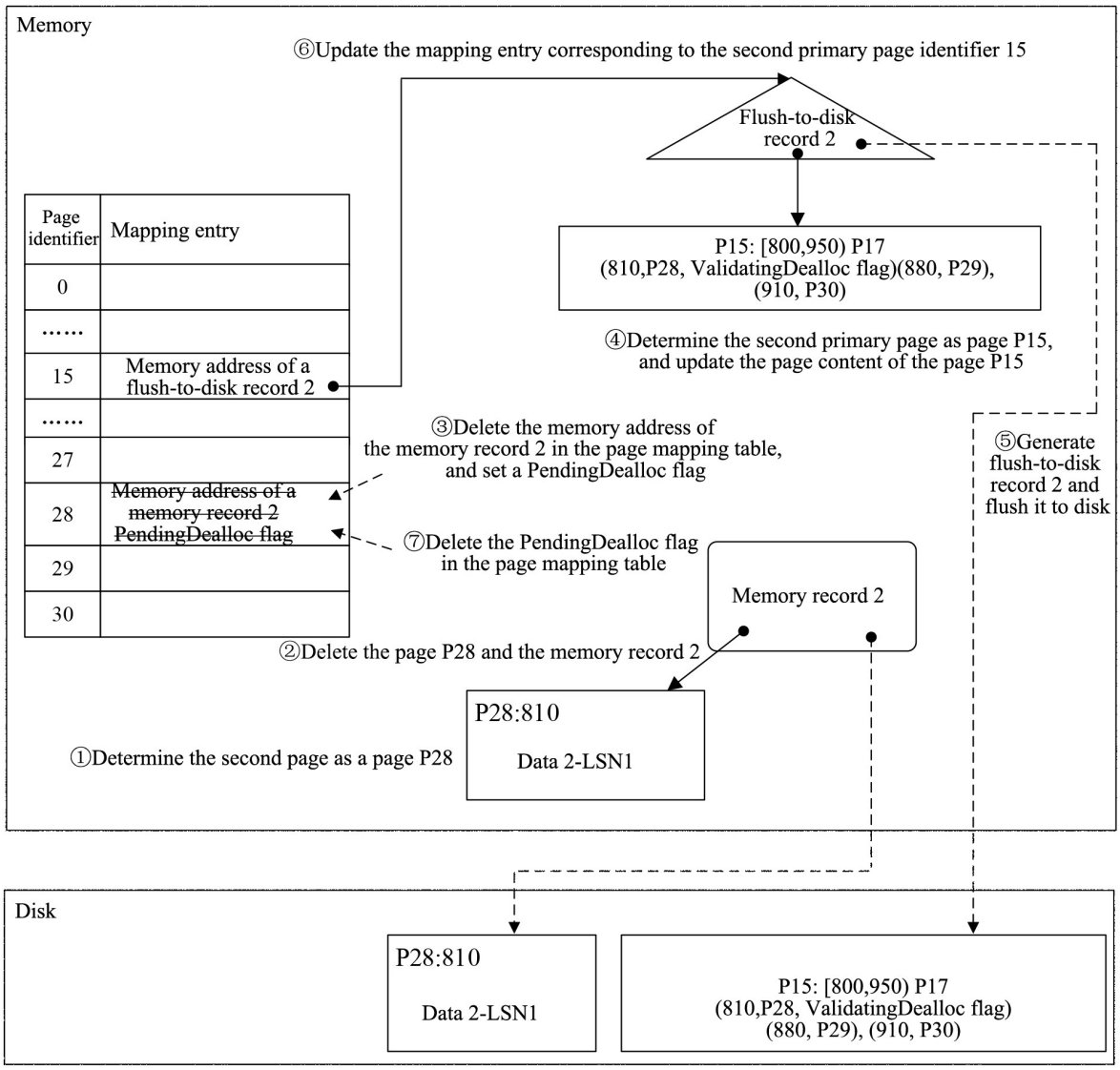

Memory
⑥Update the mapping entry corresponding to the second primary page identifier 15
Flush-to-disk record 2
Page identifier
Mapping entry
0
……
15
Memory address of a flush-to-disk record 2
……
27
28
~~Memory address of a memory record 2 PendingDealloc flag~~
29
30
P15: [800,950) P17 (810,P28, ValidatingDealloc flag)(880, P29), (910, P30)
④Determine the second primary page as page P15, and update the page content of the page P15
③Delete the memory address of the memory record 2 in the page mapping table, and set a PendingDealloc flag
⑤Generate flush-to-disk record 2 and flush it to disk
⑦Delete the PendingDealloc flag in the page mapping table
Memory record 2
②Delete the page P28 and the memory record 2
P28:810
Data 2-LSN1
①Determine the second page as a page P28
Disk
P28:810
Data 2-LSN1
P15: [800,950) P17 (810,P28, ValidatingDealloc flag) (880, P29), (910, P30)

FIG. 8

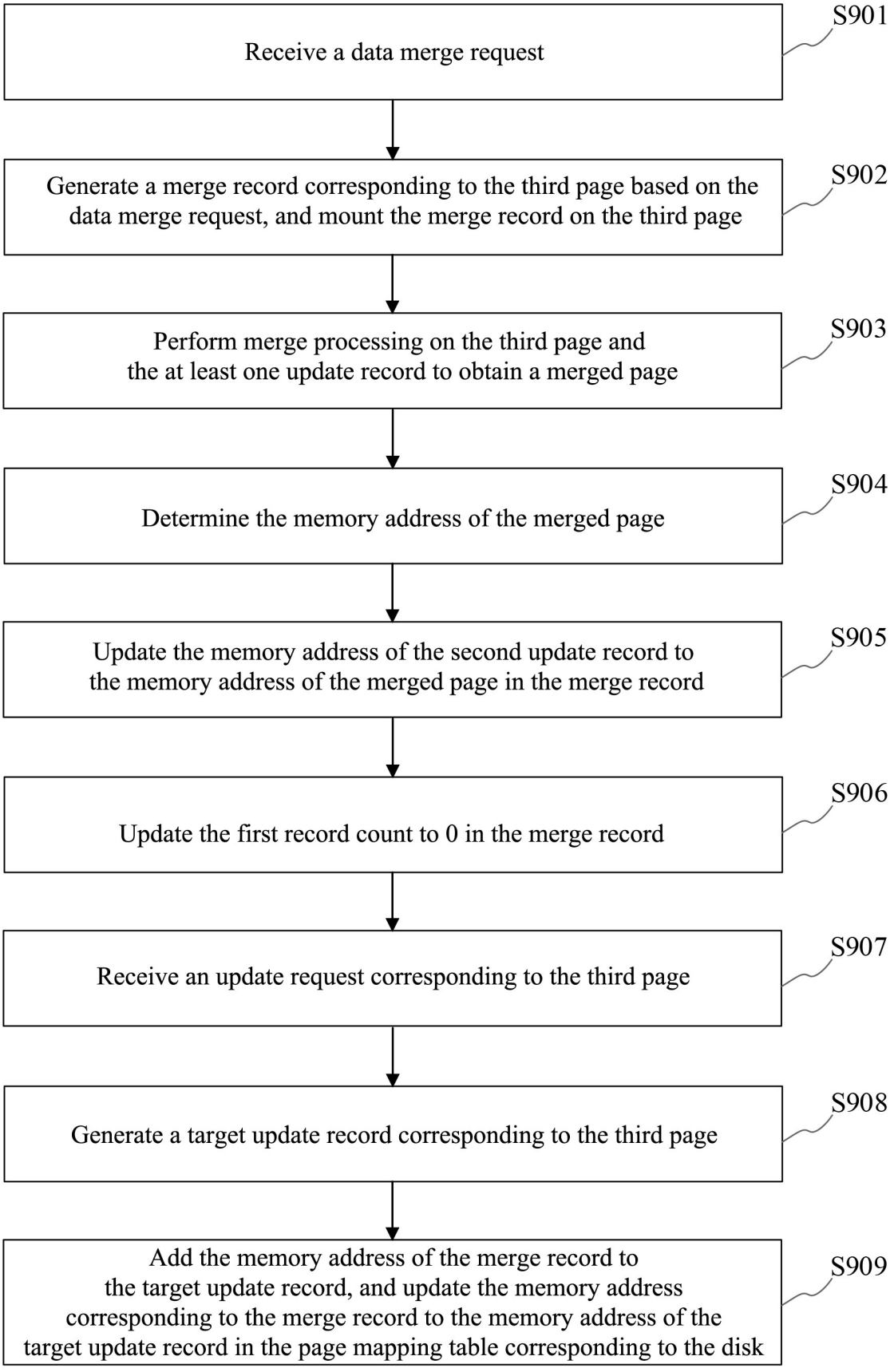

Receive a data merge request
S901
Generate a merge record corresponding to the third page based on the data merge request, and mount the merge record on the third page
S902
Perform merge processing on the third page and the at least one update record to obtain a merged page
S903
Determine the memory address of the merged page
S904
Update the memory address of the second update record to the memory address of the merged page in the merge record
S905
Update the first record count to 0 in the merge record
S906
Receive an update request corresponding to the third page
S907
Generate a target update record corresponding to the third page
S908
Add the memory address of the merge record to the target update record, and update the memory address corresponding to the merge record to the memory address of the target update record in the page mapping table corresponding to the disk
S909

FIG. 9

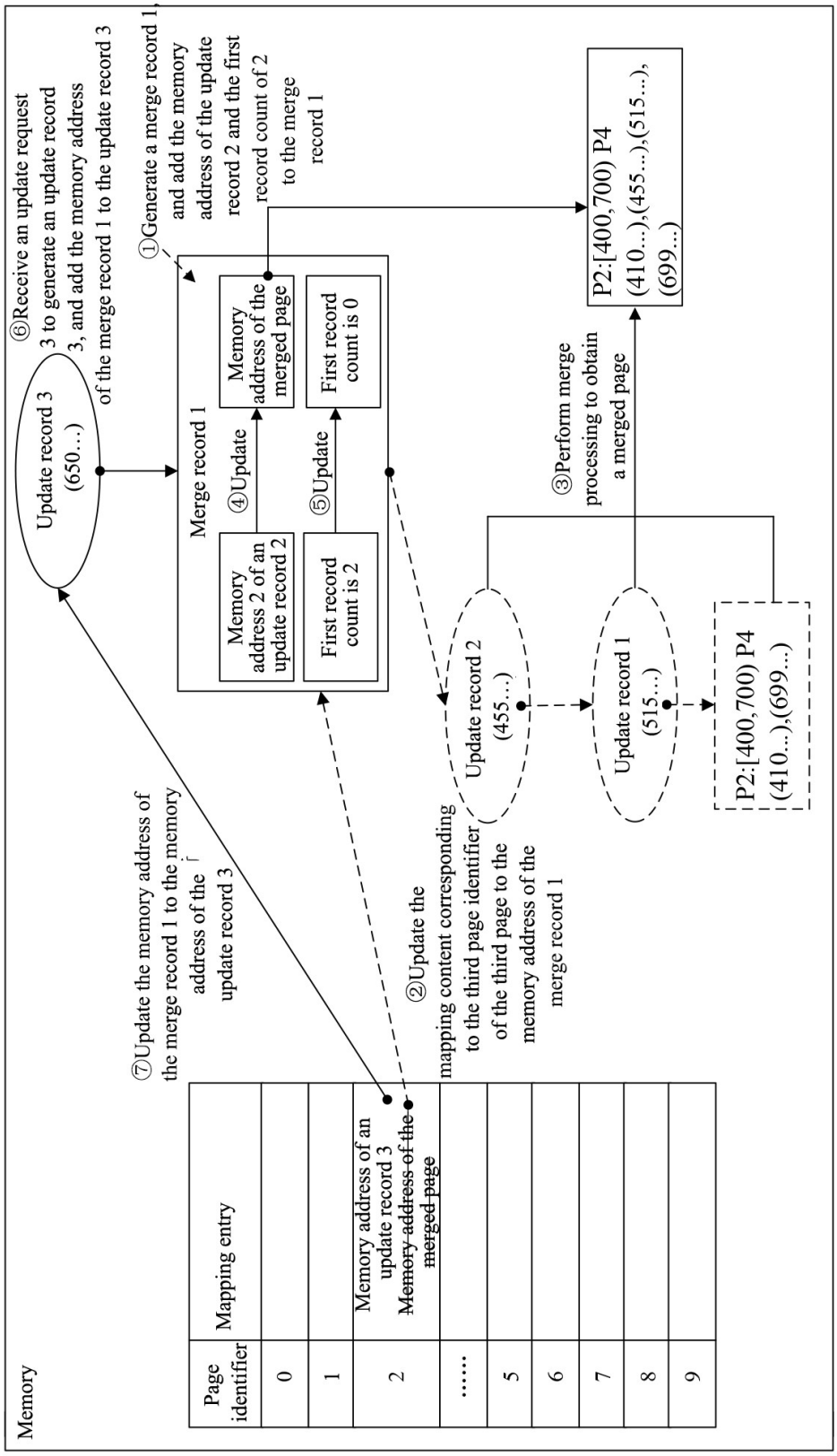

Memory
Page identifier
Mapping entry
0
1
2
.....
5
6
7
8
9
Memory address of an update record 3
~~Memory address of the merged page~~
⑥Receive an update request 3 to generate an update record 3, and add the memory address of the merge record 1 to the update record 3
Update record 3 (650…)
①Generate a merge record 1, and add the memory address of the update record 2 and the first record count of 2 to the merge record 1
Merge record 1
Memory address 2 of an update record 2
First record count is 2
④Update
⑤Update
Memory address of the merged page
First record count is 0
⑦Update the memory address of the merge record 1 to the memory address of the update record 3
②Update the mapping content corresponding to the third page identifier of the third page to the memory address of the merge record 1
Update record 2 (455…)
Update record 1 (515…)
③Perform merge processing to obtain a merged page
P2:[400,700) P4 (410...),(455...),(515...), (699...)
P2:[400,700) P4 (410...),(699...)

FIG. 10

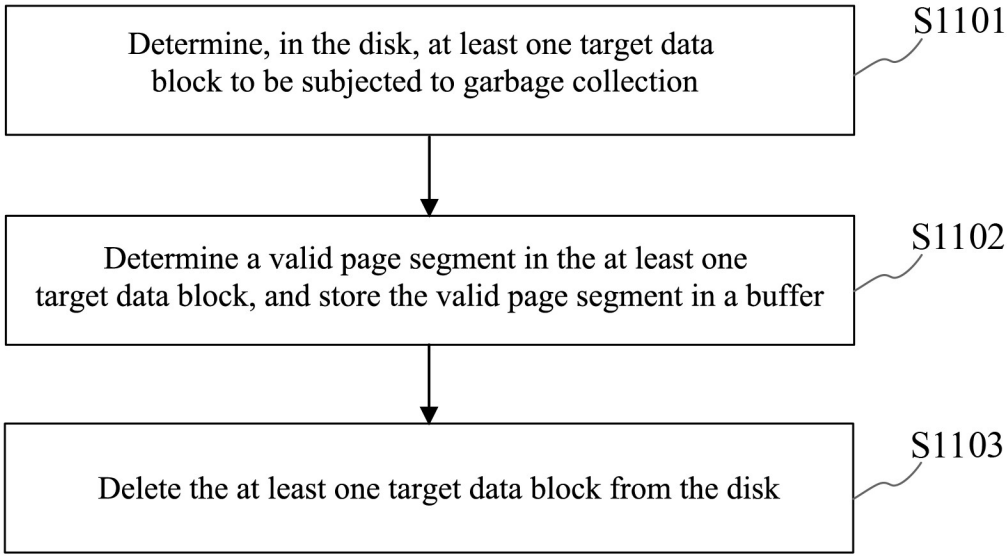

Determine, in the disk, at least one target data block to be subjected to garbage collection
S1101
Determine a valid page segment in the at least one target data block, and store the valid page segment in a buffer
S1102
Delete the at least one target data block from the disk
S1103

FIG. 11

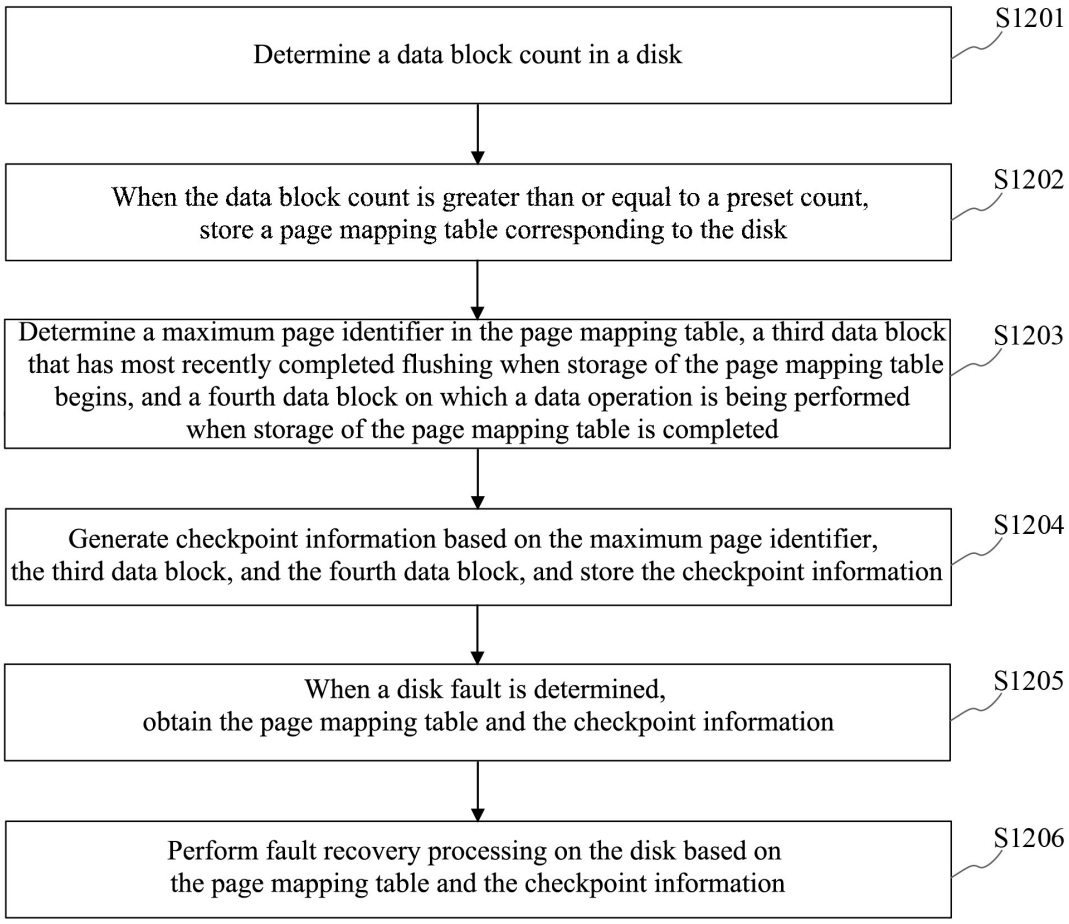

Determine a data block count in a disk
S1201
When the data block count is greater than or equal to a preset count, store a page mapping table corresponding to the disk
S1202
Determine a maximum page identifier in the page mapping table, a third data block that has most recently completed flushing when storage of the page mapping table begins, and a fourth data block on which a data operation is being performed when storage of the page mapping table is completed
S1203
Generate checkpoint information based on the maximum page identifier, the third data block, and the fourth data block, and store the checkpoint information
S1204
When a disk fault is determined, obtain the page mapping table and the checkpoint information
S1205
Perform fault recovery processing on the disk based on the page mapping table and the checkpoint information
S1206

FIG. 12

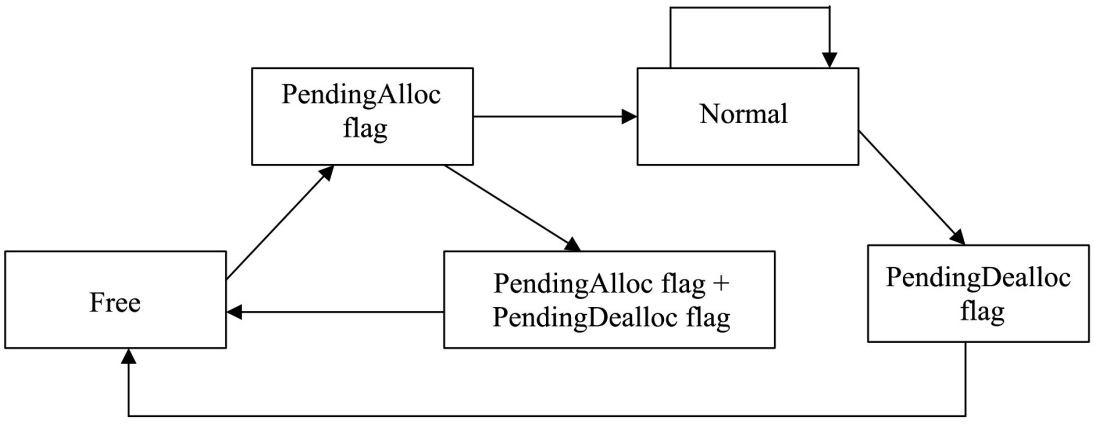

PendingAlloc flag
Normal
Free
PendingAlloc flag + PendingDealloc flag
PendingDealloc flag

FIG. 13

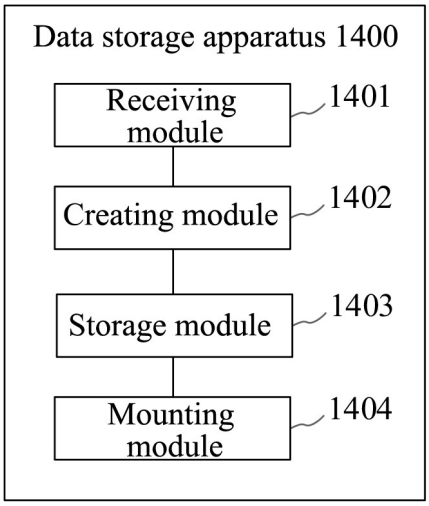

Data storage apparatus 1400
Receiving module
1401
Creating module
1402
Storage module
1403
Mounting module
1404

FIG. 14

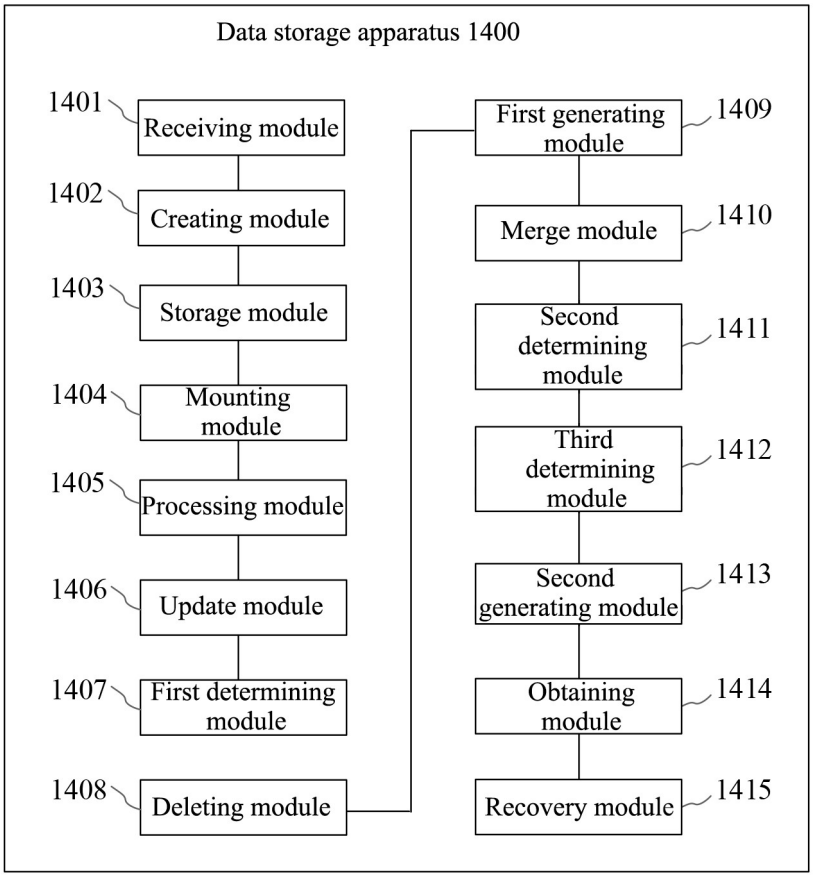

Data storage apparatus 1400
1401
Receiving module
1402
Creating module
1403
Storage module
1404
Mounting module
1405
Processing module
1406
Update module
1407
First determining module
1408
Deleting module
First generating module
1409
Merge module
1410
Second determining module
1411
Third determining module
1412
Second generating module
1413
Obtaining module
1414
Recovery module
1415

FIG. 15

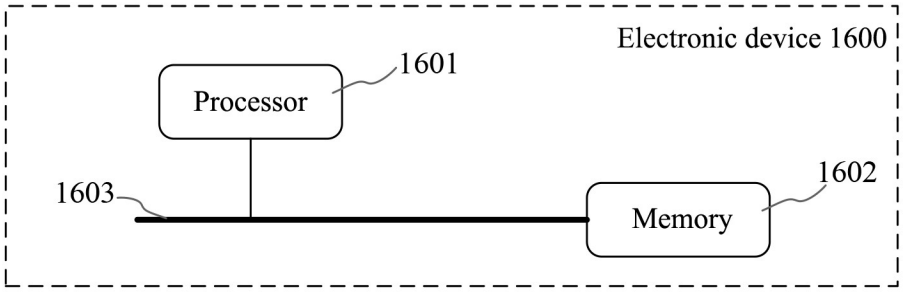


FIG. 16