(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
8 March 2007 (08.03.2007)

PCT

(10) International Publication Number
WO 2007/028026 A3

(51) International Patent Classification:
G06F 12/00 (2006.01)

(21) International Application Number:
PCT/US2006/034243

(22) International Filing Date: 31 August 2006 (31.08.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/713,913	1 September 2005 (01.09.2005)	US
11/468,569	30 August 2006 (30.08.2006)	US

(71) Applicant *(for all designated States except US)*: CYPRESS SEMICONDUCTOR CORPORATION [US/US]; 198 Champion Court, San Jose, CA 95134 (US).

(72) Inventor; and
(75) Inventor/Applicant *(for US only)*: KOLOKOWSKY, Steve [US/US]; 7915 Rufus Court, San Diego, CA 92129 (US).

(74) Agents: RICHMOND, Jeffrey, J. et al.; Marger Johnson & McCollom, P.C., 210 SW Morrison Street, Suite 400, Portland, OR 97204 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *with amended claims*

(88) Date of publication of the international search report:
27 December 2007

Date of publication of the amended claims: 6 March 2008

WO 2007/028026 A3

(54) Title: FLASH DRIVE FAST WEAR LEVELING

(57) Abstract: A system and method comprising a non- volatile memory including one or more memory blocks to store data, a controller to allocate one or more of the memor blocks to store data, and a wear-leveling table populated with pointers to unallocated memory blocks in the non-volatile memory, the controller to identify one or more pointers in the wear-leveling table and to allocate the unallocated memory blocks associated with the identified pointers for the storage of data.

AMENDED CLAIMS
[Received by the International Bureau on 07 January 2008 (07.01.08)]

1. A system comprising:

a controller to allocate one or more of the memory blocks in a non-volatile memory to store data;

a logical-to-physical table populated with pointers to the memory blocks in the non-volatile memory, the controller to initially allocate the memory blocks to store data according to the pointers in the logical-to-physical table; and

a wear-leveling table to store pointers to unallocated memory blocks in the non-volatile memory, the unallocated memory blocks previously allocated to store data by the controller according to the pointers in the logical-to-physical table, the controller to identify one or more pointers in the wear-leveling table and to store data to the memory blocks associated with the identified pointers.

2. The system of claim 1 where the controller is configured to erase data stored in one or more of the memory blocks and to populate the wear-leveling table with pointers to the erased memory blocks.

3. The system of claim 2 where the controller is configured to prioritize the pointers in the wear-leveling table according to when the memory blocks associated with the pointers were erased.

4. The system of claim 3 where the controller is configured to linearly search the wear-leveling table for pointers to unallocated memory blocks that are available to store data.

5. The system of claim 1 where the controller is configured to identify erased memory blocks and faulty memory blocks in the non-volatile memory, to compare a number of identified erased memory blocks and faulty memory blocks to a threshold, and to populate the wear-leveling table responsive to the comparing or initializing the non-volatile memory with pointers to the memory blocks responsive to the comparing.

6. The system of claim 1 where the controller initializes the logical-to-physical table with the pointers concurrently to populating the wear-leveling table with pointers to the unallocated memory blocks.

7. The system of claim 6 where the controller is configured to allocate the memory blocks associated with the pointers in the logical-to-physical table prior to the controller accessing the wear-leveling table for pointers to the unallocated memory blocks.

8. The system of claim 1

where the wear-leveling table does not store pointers to the allocated memory blocks; and

where the unallocated memory blocks were previously erased by the controller.

9. The system of claim 1 where the unallocated memory blocks include one or more memory blocks available to store data and one or more faulty memory blocks not to store data.

10. The system of claim 9 where the wear-leveling table is configured to indicate to the controller which of the pointers are associated with the faulty memory blocks.

11. The system of claim 1 where the non-volatile memory includes multiple memory zones, each memory zone including a plurality of memory blocks and having a different wear-leveling table, the controller to receive a write command identifying a memory zone to store data, to determine the wear-leveling table does not correspond to the identified memory zone, and to re-populate the wear-leveling table with pointers to memory blocks corresponding to the identified memory zone.

12. A method comprising:

populating a wear-leveling table with pointers to one or more memory blocks in a flash memory, where the pointers are associated with erased memory blocks or memory blocks that are faulty;

identifying at least one pointer in the wear-leveling table associated with an erased memory block;

storing data to the memory block associated with the identified pointer;

identifying erased memory blocks and faulty memory blocks in the flash memory;

comparing a number of identified erased memory blocks and faulty memory blocks to a threshold; and

populating the wear-leveling table responsive to the comparing or initializing the flash memory with pointers to the memory blocks responsive to the comparing.

13. The method of claim 12 includes erasing data stored in one or more of the memory blocks responsive to the identifying at least one pointer in the wear-leveling table and prior to storing data to the memory block associated with the identified pointer.

14. The method of claim 12 includes prioritizing the pointers in the wear-leveling table according to when the memory blocks associated with the pointers were erased.

15. The method of claim 14 includes accessing the wear-leveling table for pointers according to the prioritizing.

16. The method of claim 12 includes prioritizing the pointers in the wear-leveling table according to the number of write operations performed on the memory blocks associated with the pointers.

17. The method of claim 12 includes initializing a logical-to-physical table with pointers to the memory blocks in the flash memory concurrently to populating the wear-leveling table with pointers associated with the erased memory blocks.

18. The method of claim 17 includes allocating the memory blocks associated with the pointers in the logical-to-physical table prior to accessing the wear-leveling table for pointers to erased memory blocks.

19. A method comprising:

populating a wear-leveling table with pointers to one or more memory blocks in a flash memory, where the pointers are associated with erased memory blocks or memory blocks that are faulty;

receiving a write command identifying a zone of the flash memory to store data;

determining the wear-leveling table does not correspond to the identified zone;

re-populating the wear-leveling table with pointers to memory blocks corresponding to the identified zone;

identifying at least one pointer in the re-populated wear-leveling table associated with an erased memory block; and

storing data to the memory block associated with the identified pointer.

20. The method of claim 19

identifying erased memory blocks and faulty memory blocks in the flash memory;

comparing a number of identified erased memory blocks and faulty memory blocks to a threshold; and

populating the wear-leveling table responsive to the comparing.

21. The method of claim 19

identifying erased memory blocks and faulty memory blocks in the flash memory;

comparing a number of identified erased memory blocks and faulty memory blocks to a threshold; and

initializing the flash memory with pointers to the memory blocks responsive to the comparing.