US 20110271043A1

(19) United States

(12) Patent Application Publication

(10) Pub. No.: US 2011/0271043 A1

(43) Pub. Date: Nov. 3, 2011

SEGAL et al.

(54) SYSTEM AND METHOD FOR ALLOCATING AND USING SPARE BLOCKS IN A FLASH MEMORY

(76) Inventors: **Avigdor SEGAL**, Netanya (IL); **Igal Maly**, Tel Aviv (IL)

(21) Appl. No.: **13/096,736**

(22) Filed: **Apr. 28, 2011**

Related U.S. Application Data

(60) Provisional application No. 61/329,366, filed on Apr. 29, 2010.

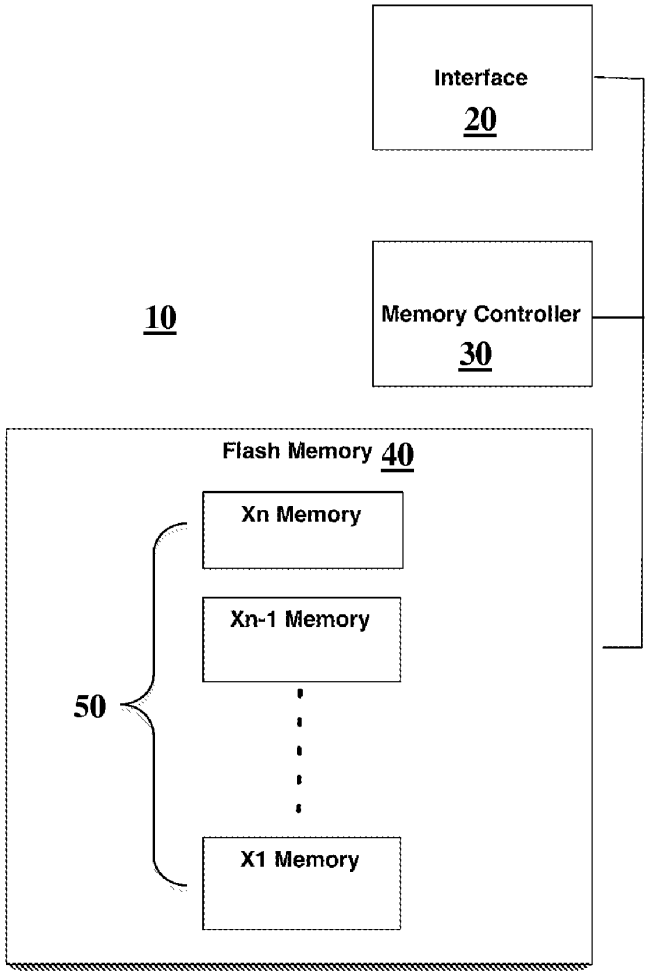
Publication Classification

(51) **Int. Cl.**
G06F 12/00 (2006.01)

(52) **U.S. Cl.** **711/103**; 711/E12.001

(57) **ABSTRACT**

A method for using a single spare block pool in flash memory comprising: allocating a plurality of flash memory arrays, wherein each flash memory array comprises a plurality of flash memory blocks; within a main flash memory array: allocating a used block pool comprising a plurality of used blocks and allocating a main spare block pool comprising a plurality of spare blocks; within each of the other flash memory arrays: allocating a used block pool comprising multiple used blocks; allocating a minimum spare block pool comprising a minimum number of spare blocks; allocating the main spare block pool and each of the minimum spare block pools to a single spare block pool; transferring a spare block from the main spare block pool to one of the minimum spare block pools; and transferring a spare block from a first minimum spare block pool to a second minimum spare block pool.



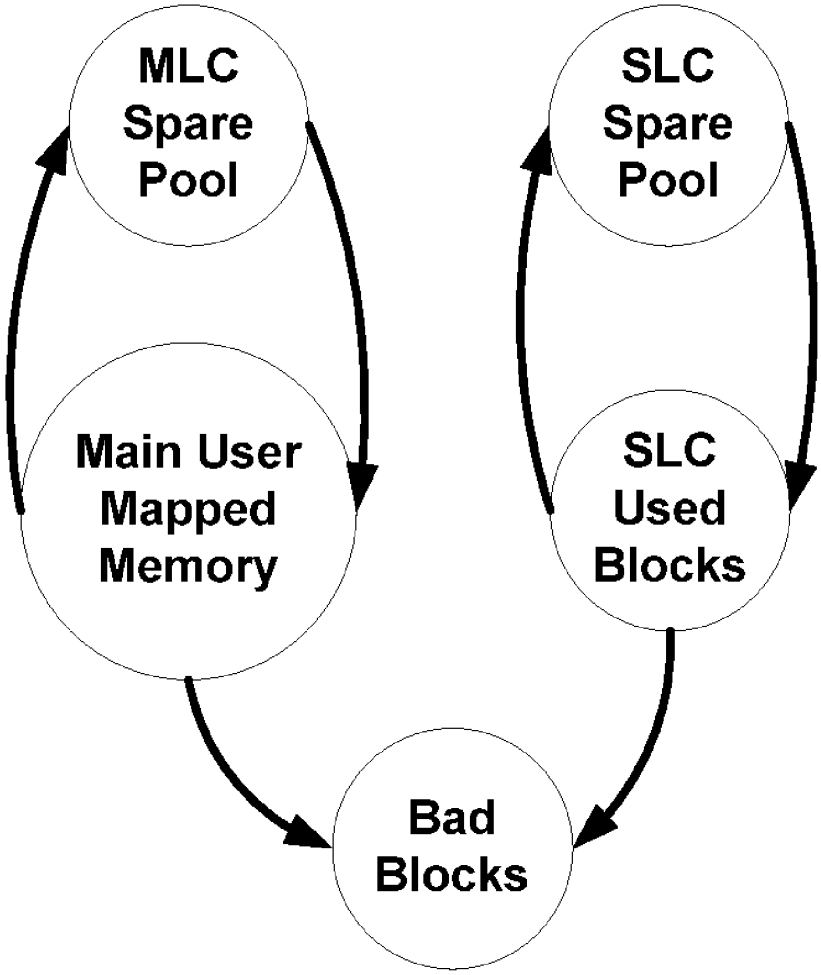

MLC
Spare
Pool
SLC
Spare
Pool
Main User
Mapped
Memory
SLC
Used
Blocks
Bad
Blocks

FIG. 1

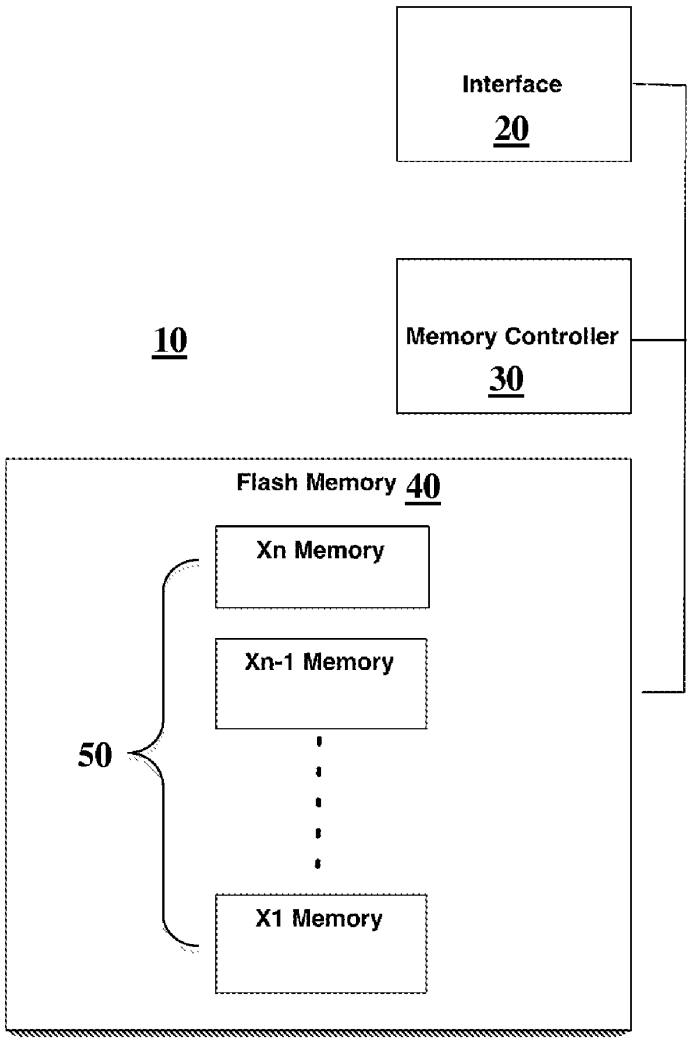

Interface
20
Memory Controller
30
10
Flash Memory 40
Xn Memory
Xn-1 Memory
50
X1 Memory

FIG. 2

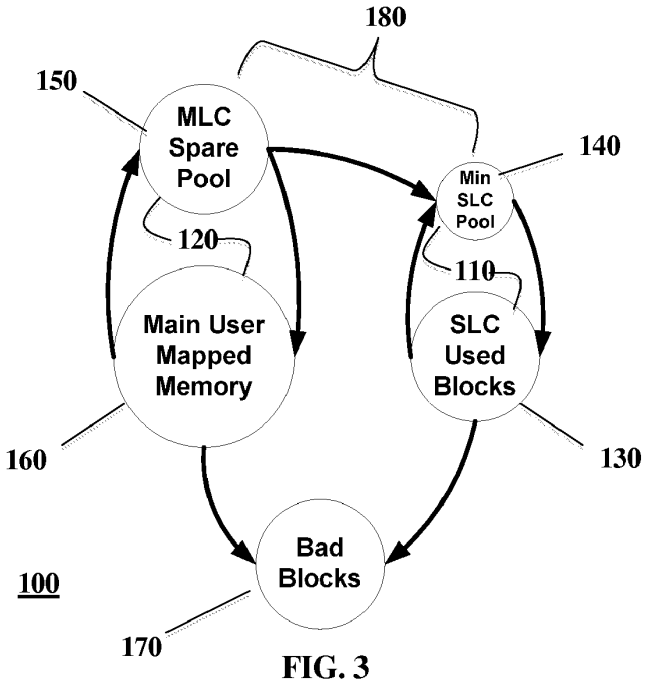

180
150
MLC Spare Pool
140
Min SLC Pool
120
110
Main User Mapped Memory
SLC Used Blocks
160
130
Bad Blocks
100
170

FIG. 3

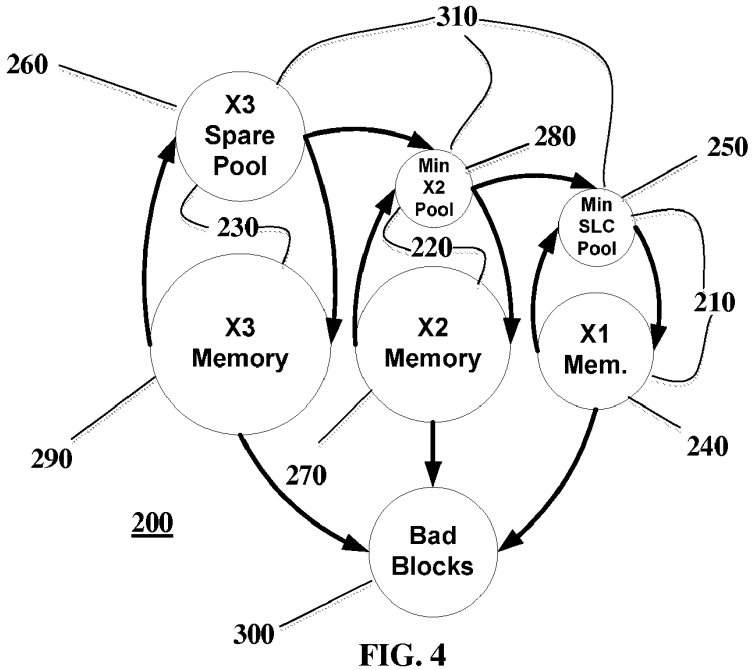

310
260
X3 Spare Pool
280
250
Min X2 Pool
Min SLC Pool
230
220
210
X3 Memory
X2 Memory
X1 Mem.
290
270
240
200
Bad Blocks
300

FIG. 4

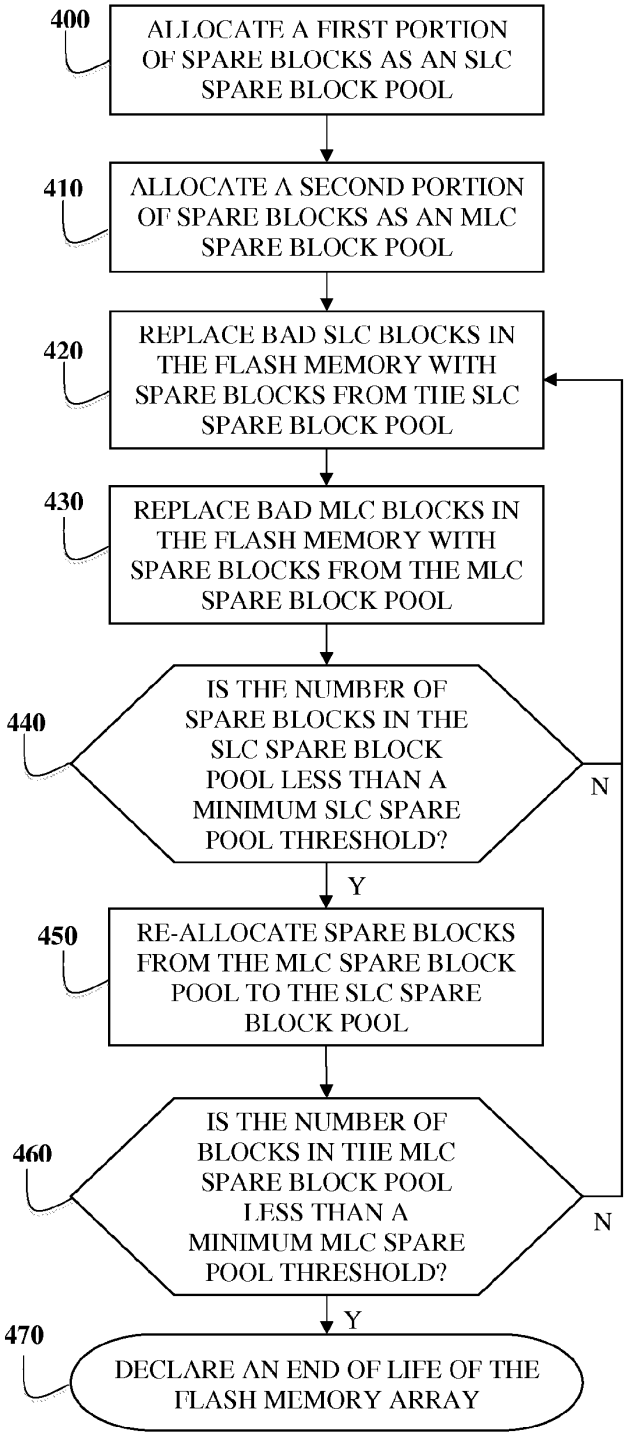

400
ALLOCATE A FIRST PORTION OF SPARE BLOCKS AS AN SLC SPARE BLOCK POOL
410
ALLOCATE A SECOND PORTION OF SPARE BLOCKS AS AN MLC SPARE BLOCK POOL
420
REPLACE BAD SLC BLOCKS IN THE FLASH MEMORY WITH SPARE BLOCKS FROM THE SLC SPARE BLOCK POOL
430
REPLACE BAD MLC BLOCKS IN THE FLASH MEMORY WITH SPARE BLOCKS FROM THE MLC SPARE BLOCK POOL
440
IS THE NUMBER OF SPARE BLOCKS IN THE SLC SPARE BLOCK POOL LESS THAN A MINIMUM SLC SPARE POOL THRESHOLD?
N
Y
450
RE-ALLOCATE SPARE BLOCKS FROM THE MLC SPARE BLOCK POOL TO THE SLC SPARE BLOCK POOL
460
IS THE NUMBER OF BLOCKS IN THE MLC SPARE BLOCK POOL LESS THAN A MINIMUM MLC SPARE POOL THRESHOLD?
N
Y
470
DECLARE AN END OF LIFE OF THE FLASH MEMORY ARRAY

FIG. 5

SYSTEM AND METHOD FOR ALLOCATING AND USING SPARE BLOCKS IN A FLASH MEMORY

CROSS-REFERENCE TO RELATED APPLICATION

[0001] This application claims the benefit of U.S. Provisional Patent Application No. 61/329,366, filed Apr. 29, 2010, which is incorporated herein by reference in its entirety.

FIELD OF THE INVENTION

[0002] Embodiments of the present invention relate generally to systems and methods involving multi-level non-volatile memory, and particularly to allocating and/or using a spare block pool in such a memory device.

BACKGROUND OF THE INVENTION

[0003] Flash memory devices, which include arrays of Flash memory cells, are used in many applications for storing digital information. Flash memory is frequently used in portable electronic devices, digital cameras, personal computers, memory cards, and other types of devices. Due to the nature of these devices, the endurance, speed, and longevity of the Flash memory are important.

[0004] Flash memory devices may store data in arrays of Flash memory cells, including single-level and/or multi-level cells. A single level cell (SLC) flash memory stores one bit of information per cell, typically by programming and reading a binary charge (e.g., high or low charge). In contrast, multi-level cells (MLC) store multiple bits of information in each cell by programming and reading a level of charge in the cell. Multi-level cells may include cells that store two bits (MLCx2), three bits (MLCx3), N bits (MLCxN), etc. Flash memory cells are organized into blocks within the array, and the blocks may be arranged by block type into flash memory arrays, or populations of block types.

[0005] The majority of blocks in a Flash memory device are usable blocks, or good blocks. However, inevitably, every Flash device includes non-functional blocks, commonly referred to as bad blocks, which may be incapable of being erased or rewritten with new information for any number of reasons. Typically, a Flash memory device may include some bad blocks at the beginning of the life of the device, and the number of bad blocks typically increases during the lifetime of the device. Flash memory blocks can become bad from any number of reasons. For example, each block of flash memory is limited in the number of times the data therein can be programmed and erased, characterized by a maximum program/erase (P/E) cycle. Thus, if a block of the Flash memory has been programmed and erased a number of times exceeding the P/E cycle maximum, it may turn bad. Flash memory blocks may also become bad blocks by erase failure, or other causes.

[0006] In most instances, the capacity of a memory card or other device using Flash memory (referred to generally as a Flash device) must be guaranteed during a rated lifetime of the device. The Flash device capacity may be directly limited by the number of usable blocks available to store data. Therefore, in order to ensure that the card capacity is maintained throughout the rated lifetime of the Flash device, a system designer may allocate a certain amount of usable blocks in a spare pool to allow switching of good spare blocks instead of bad used blocks during usage. Maintaining a pool of spare blocks from which good blocks may be drawn upon in exchange for bad blocks allows a requisite number of usable blocks to be ensured throughout the life of the device.

[0007] In MLC flash memory devices, memory blocks may be used as different types, for example SLC, MLCx2, MLCx3 blocks, etc., each of which may have different reliability specifications. Due to the differing reliability specifications of each data block type, a designer may allocate different numbers of spare blocks associated with each block type.

[0008] FIG. **1** illustrates a Flash memory device with a known method of allocating and using spare blocks. A spare block pool of single level cells (an SLC spare block pool) exchanges blocks exclusively with an SLC used block pool. A spare block pool of multi-level cells (a MLC spare block pool) exchanges blocks exclusively with a main user mapped memory of used MLC blocks. Bad blocks are exchanged for spare blocks, and the flash memory device can no longer write information to the bad blocks. The pools are set at the start of the life of the flash memory device. Since the usage of each data type is unknown, the maximum spare blocks are typically allocated to each of the SLC spare block pool and the MLC spare block pool. The solution in this case is however sub-optimal because at the end of the lifetime of the memory device, some of the spare blocks from either the SLC spare block pool or MLC spare block pool will typically not be used.

[0009] However, the rate of deterioration may be different for the different block types, such that when a spare block pool of one type is depleted, the memory device may reach the end of its usable life, even though there may be spare blocks of other types remaining. For example, at a certain point in the life of the device, there may be too few usable SLC blocks to maintain the specified flash memory card capacity while a surplus of MLC block types remain. The lifetime of the device will therefore end while usable MLC spare blocks are still available. Consequently, all of the flash memory blocks will be sub-optimally exploited during the lifespan of the device.

[0010] A similar problem arises when handling caching and buffering for the flash memory write operation or when executing the static and dynamic wear leveling processes. Buffer and cache operations and static and dynamic wear leveling processes may both require use of spare blocks. A problem may arise when there are too few good blocks of one of the block populations, e.g., SLC, MLC, to allocate for buffering and caching, or for static and dynamic wear leveling.

[0011] For the foregoing reasons, there is a need for a flash memory method that ensures optimal usage of all blocks throughout the life of the flash memory device and increases longer flash memory device lifespan.

SUMMARY OF EMBODIMENTS OF THE INVENTION

[0012] Embodiments of the present invention provide systems and methods for using Flash memory by allocating a first portion of spare blocks in the Flash memory as a single level cell (SLC) spare block pool, and allocating a second portion of spare blocks in the Flash memory as a multi-level cell (MLC) spare block pool. Bad SLC blocks in the Flash memory may be replaced with spare blocks from the SLC spare block pool. If the number of spare blocks in the SLC spare block pool is less than a minimum SLC spare pool

threshold, then spare blocks from the MLC spare block pool may be re-allocated to the SLC spare block pool.

[0013] Embodiments of the present invention provide systems and methods for using non-volatile memory by allocating N number of portions of spare blocks in the non-volatile memory as N number of multi-level cell (MLCx(N)) spare block pools. Bad MLCx(j) blocks in the non-volatile memory may be exchanged with spare blocks from an MLCx(j) spare block pool. If the number of spare blocks in an MLCx(j) spare block pool is less than a minimum MLCx(j) spare pool threshold, spare blocks may be transferred from the MLCx(j+1) spare block pool to the MLCx(j) spare block pool.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] The subject matter regarded as the invention is particularly pointed out and distinctly claimed in the concluding portion of the specification. The invention, however, both as to organization and method of operation, together with objects, features, and advantages thereof, may best be understood by reference to the following detailed description when read with the accompanying drawings in which:

[0015] FIG. **1** schematically illustrates a prior art method for allocating and using spare pools of MLC and SLC blocks;

[0016] FIG. **2** schematically illustrates a structure of a Flash memory device according to an embodiment of the present invention;

[0017] FIG. **3** illustrates a method for allocating and using spare pools of MLC and SLC blocks in accordance with an embodiment of the present invention having one type of MLC;

[0018] FIG. **4** illustrates a method for allocating and using spare pools of MLC and SLC blocks in accordance with an embodiment of the present invention having two types of MLC; and

[0019] FIG. **5** is a flowchart of a method for allocating and using spare pools of MLC and SLC blocks according to an embodiment of the present invention.

[0020] It will be appreciated that for simplicity and clarity of illustration, elements shown in the figures have not necessarily been drawn to scale. For example, the dimensions of some of the elements may be exaggerated relative to other elements for clarity. Further, where considered appropriate, reference numerals may be repeated among the figures to indicate corresponding or analogous elements.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

[0021] In the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the invention. However, it will be understood by those skilled in the art that the present invention may be practiced without these specific details. In other instances, well-known methods, procedures, and components, modules, units and/or circuits have not been described in detail so as not to obscure the invention.

[0022] Although embodiments of the invention are not limited in this regard, discussions utilizing terms such as, for example, "processing," "computing," "calculating," "determining," "establishing", "analyzing", "checking", or the like, may refer to operation(s) and/or process(es) of a computer, a computing platform, a computing system, or other electronic computing device, that manipulates and/or transforms data represented as physical (e.g., electronic) quantities within the computer's registers and/or memories into other data similarly represented as physical quantities within the computer's registers and/or memories or other information non-transitory storage medium that may store instructions to perform operations and/or processes.

[0023] Although embodiments of the invention are not limited in this regard, the terms "plurality" and "a plurality" as used herein may include, for example, "multiple" or "two or more". The terms "plurality" or "a plurality" may be used throughout the specification to describe two or more components, devices, elements, units, parameters, or the like. Unless explicitly stated, the method embodiments described herein are not constrained to a particular order or sequence. Additionally, some of the described method embodiments or elements thereof can occur or be performed simultaneously, at the same point in time, or concurrently.

[0024] According to embodiments of the present invention, a common spare block pool may be used in a Flash memory so as to increase the usable lifetime of a Flash memory device by more fully exploiting spare memory blocks.

[0025] FIG. **2** schematically illustrates a flash memory device **10**. The flash memory device comprises an interface **20**, a memory controller **30**, and a flash memory module **40**. The flash memory module comprises multiple (n) memory arrays **50**. The memory controller **30** manages the reading, writing, erasing, and location of data stored in the flash memory module **40**. The memory controller **30** also manages the exchange of flash memory blocks between different pools of flash memory blocks in the flash memory module **40**. The memory controller **30** also controls buffering and caching operations; bad block management; and static and dynamic wear leveling processes within the flash memory module **40**. It will be recognized that in some embodiments of the invention, the memory controller **30** may be integrated into the flash memory module.

[0026] According to embodiments of the invention, upon initialization of the Flash memory, a common pool of spare blocks may be apportioned as a pool of spare SLC blocks and a pool of spare MLC blocks. More specifically, in some embodiments of the invention, an initial number of the spare blocks may initially be allocated as a pool of spare SLC blocks, and the remainder of the spare blocks may be allocated as a pool of spare MLC blocks. The initial number of spare blocks allocated for use as a spare pool of SLC blocks may be a small number, for example, the minimum number of spare blocks required for smooth operation of the SLC memory cells, for example, two spare blocks. The remainder of the spare blocks may be allocated for use as a spare pool of MLC blocks.

[0027] As described below, during use of the Flash device, memory blocks in use are periodically tested to identify bad blocks, and bad blocks are exchanged with spare blocks from the spare block pool of the corresponding type. According to embodiments of the invention, upon reaching below a minimum level of spare SLC blocks, spare blocks allocated to the MLC spare block pool may be transferred to the SLC spare block pool to maintain a minimum number of spare blocks. Additionally or alternatively, according to embodiments of the invention, MLC blocks in use may be transferred to the SLC spare block pool upon reaching a use condition, such as a maximum MLC bearable cycle count. SLC blocks that reach a maximum cycle count may be declared bad blocks. When no more MLC spare blocks are available, the Flash device may end its lifetime.

[0028] FIG. **3** schematically illustrates the functional processing of blocks in an MLC Flash memory device **100** according to an embodiment of the present invention having one type of MLC. It will be understood that FIG. **3** is a schematic illustration, and that blocks are not physically moved between or among pools, but rather various data structures, e.g., look-up tables, pointers, etc., which may be used to identify spare blocks of various types, blocks in use, and bad blocks, etc., may be modified as if to move or exchange an association of various blocks from spare block to block in use, or from block in use to bad block, etc.

[0029] As shown in FIG. **3**, Flash memory device **100** may include an SLC flash memory array **110** and an MLC flash memory array **120**. The SLC flash memory array **110** comprises an SLC used memory pool **130** and a number of spare blocks allocated as a pool of SLC spare blocks **140**. As mentioned above, in an embodiment of the invention, SLC spare block pool **140** may initially be determined as a minimum number of spare blocks required for smooth functioning of the SLC memory array, such that only the minimum requisite spare blocks required for the SLC memory array **110** to function according to the SLC array **110** and MLC flash memory device **100** specifications are initially provided. The MLC flash memory array **120** also comprises an MLC used block pool **160** and an MLC spare block pool **150**. The remaining spare blocks, e.g., those spare blocks not assigned to the SLC spare block pool may be allocated to the MLC spare block pool **150**.

[0030] For example, in an embodiment of the invention, the SLC flash memory array **110**, may require 10 usable SLC blocks and one spare SLC block at any arbitrary moment for smooth operation. A total of 11 SLC blocks may therefore be allocated to the SLC flash memory array **110** at system initialization, ten blocks of which are for general use in the SLC memory pool **130**, and one block of which is allocated to the SLC spare blocks pool **140**. The remaining blocks are allocated to the MLC spare block pool **150**.

[0031] According to an embodiment of the invention, during normal operation of the flash memory device **100**, the memory controller **30** may scan and test the flash memory arrays to identify bad blocks **170**. Various methods are known for testing for bad blocks, which are not described herein for purposes of brevity. If a bad block **170** is discovered in the MLC memory array **120**, the MLC bad block may be exchanged with a spare block from the MLC spare block pool **150**. If a bad block is discovered in the SLC flash memory array **110**, the bad block may be exchanged with a spare block from the SLC spare block pool **140**.

[0032] In some embodiments of the invention, the memory device may require a predetermined minimum number of spare SLC blocks to be available at any time. Therefore, according to embodiments of the invention, when, for example, due to usage of SLC spare blocks, the number of available spare SLC blocks falls below this minimum, spare blocks from the MLC spare pool may be transferred, or re-allocated for use as SLC spare blocks as required.

[0033] In an embodiment of the invention, dynamic and static wear leveling operations may be performed within the SLC memory array **110** and the MLC memory array **120**. Dynamic and static wear leveling operations may be performed, for example, in order to extend the useful life of the Flash memory device. Each flash memory block is typically rated for a certain number of P/E cycles. For example, MLC blocks may be rated for 3,000 P/E cycles and SLC blocks may be rated for 50,000 P/E cycles. Dynamic and static wear leveling may prevent or delay some blocks of memory from exceeding a predetermined P/E cycle threshold while other blocks have endured fewer P/E cycles by distributing Flash cell erasures and re-writes evenly across the Flash memory array.

[0034] In an embodiment of the invention, MLC blocks from the MLC memory array **120** that reach a predetermined maximum MLC bearable P/E cycles may be transferred to the SLC spare block pool **140** and called into use as SLC blocks as required. In some embodiments of the invention, a maximum number of bearable P/E cycles for an MLC block may be approximately 3,000 P/E cycles; a maximum number of bearable P/E cycles for an SLC block may be approximately 50,000 P/E cycles. According to embodiments of the invention, MLC blocks, therefore, may be used as SLC blocks after exceeding the maximum P/E cycle rating for an MLC block but while still less than the maximum P/E cycle rating for SLC blocks. SLC blocks that reach the maximum P/E cycle count for an SLC block may be declared bad blocks. Accordingly, the end of the Flash memory device's life may be declared when the number of remaining MLC spare blocks is zero. In some embodiments of the invention, when a device reaches the end of its lifetime, no more writes may be allowed, and the device may be declared Read-Only.

[0035] FIG. **4** schematically illustrates an embodiment of the invention having three populations of block types, e.g., MLCx3, MLCx2, and SLC. Memory device **200** comprises an SLC memory array **210**, a first MLC memory array MLCx2 **220**, and a second MLC memory array MLCx3 **230**. The SLC flash memory array **210** comprises an SLC used memory pool **240** and a small number of spare blocks initially allocated to a SLC spare block pool **250**. In some embodiments of the invention, a minimum number of spare blocks required for the smooth operation of SLC memory array **210** may be initially provided. MLCx2 memory array **220** comprises an MLCx2 used block pool **270** and a small number of spare blocks initially allocated to an MLCx2 spare block pool **280**. In some embodiments of the invention, a minimum number of spare blocks required for the smooth operation of MLCx2 memory array **220** may be initially provided. As discussed below, because the spare blocks in the MLCx2 spare block pool **280** may be used as MLCx2 spare blocks and/or SLC spare blocks, the MLCx2 spare block pool **280** may function as a common spare block pool for the MLCx2 and SLC blocks in use. The remaining spare blocks may be allocated to MLCx3 spare block pool **260**. As discussed below, because the spare blocks in the MLCx3 spare block pool **260** may be used as MLCx3 spare blocks, MLCx2 spare blocks, and/or SLC spare blocks, the MLCx3 may function as a common spare block pool for the MLCx3, MLCx2 and SLC blocks in use.

[0036] During normal operation of Flash memory device **200**, memory controller **30** may regularly scan and test the memory arrays to identify bad blocks, e.g., a bad SLC block is discovered, or a SLC memory block has exceeded the prescribed maximum number of bearable P/E cycles. In an embodiment of the invention, bad SLC blocks in use in SLC blocks **240** may be exchanged for good SLC blocks from SLC spare block pool **250**. Bad MLCx2 blocks in use in MLCx2 blocks **270** may be exchanged for good MLCx2 blocks from MLCx2 spare block pool **280**. Bad MLCx3 blocks in use in MLCx3 blocks **290** may be exchanged for good MLCx3 blocks from MLCx2 spare block pool **260**.

[0037] As mentioned above, the number of SLC blocks allocated to the SLC spare block pool may be small, and therefore, the pool may be depleted of spare blocks after operation of SLC blocks in the device. Accordingly, if SLC blocks are no longer available in the SLC spare block pool, or if the number of spare blocks in the SLC spare block pool falls below a minimum threshold, e.g., a minimum number required for smooth operation of the SLC memory, a spare block from the MLCx2 spare block pool may be reallocated for the SLC spare block pool.

[0038] Likewise, the number of MLCx2 blocks allocated to the MLCx2 spare block pool may be small, and therefore, the pool may be depleted of spare blocks after operation of MLCx2 blocks in the device. Accordingly, if MLCx2 blocks are no longer available in the MLCx2 spare block pool, or if the number of spare blocks in the MLCx2 spare block pool falls below a minimum threshold, e.g., a minimum number required for smooth operation of the MLCx2 memory, a spare block from the MLCx3 spare block pool may be reallocated for the MLCx2 spare block pool.

[0039] In an embodiment of the invention, dynamic and static wear leveling operations may be performed in any one or more of the SLC memory array **210**, the MLCx2 memory array **220**, and the MLCx3 memory array **230** in order to increase the lifetime of the flash memory.

[0040] Additionally or alternatively, in an embodiment of the invention, MLCx3 blocks in use from the MLCx3 memory array **230** that reach maximum bearable cycle count for MLCx3 memory may be transferred to the MLCx2 spare block pool **280**. Similarly, MLCx2 blocks in the MLCx2 memory array **220** that reach maximum bearable cycle count specified for MLCx2 memory may be transferred to SLC spare block pool **250**. SLC blocks that reach the maximum bearable P/E cycle count for an SLC block may be declared bad blocks **300**.

[0041] According to an embodiment of the invention, the end of the Flash memory device **200** life may be declared to occur when the number of MLCx3 spare blocks is less than a predefined minimum spare blocks required for the X3 flash memory device **200** to operate according to specification.

[0042] It will be understood that the above example involving one type of MLC block, e.g., an MLCx2, and SLC blocks may be generalized to include many types of MLC, e.g., MLCxN, MLCx(N−1) . . . MLCx2 and SLC blocks. In such embodiments of the invention, each of the lower levels of cell blocks (SLC, MLCx2 . . . MLCx(N−1)) may be allotted the minimum number of spare blocks, and upon a type of spare block pool being depleted of spare blocks, additional spare blocks may be allocated from the spare block pool of a higher-order MLC block pool. Additionally or alternatively, upon exhaustion of an MLC block in use by reaching a maximum number of bearable P/E cycles for that type of MLC block, e.g., MLCx(j), the block may be reallocated as a spare block of type MLCx(j−1). It will be understood that in an SLC block may be considered analogous to MLCx1.

[0043] FIG. **5** is a flowchart of a method for allocating and using spare pools of MLC and SLC blocks in an MLC Flash memory device **100** according to an embodiment of the present invention.

[0044] In operation **400**, a controller (e.g., controller **30** of FIG. **2**) may allocate a first portion of spare blocks as an SLC spare block pool (e.g., SLC spare block pool **140** of FIG. **3**). The controller may, in some embodiments, allocate a minimum number of spare blocks needed for smooth operation of the SLC memory cells (e.g., SLC memory array **110** of FIG. **3**) and/or the Flash memory device (e.g., Flash memory device **100** of FIG. **1**).

[0045] In operation **410**, the controller may allocate a second portion of spare blocks as an MLC spare block pool (e.g., MLC spare block pool **150** of FIG. **3**). The controller may, in some embodiments, allocate to the MLC spare block pool all spare blocks not assigned to the SLC spare block pool.

[0046] In operation **420**, the controller may replace bad SLC blocks in the Flash memory with spare blocks from the SLC spare block pool. An SLC block or other type of Flash memory block may become a bad block, for example, if the block is programmed and erased a number of times exceeding the P/E cycle maximum for the SLC block. Flash memory blocks may also become bad blocks by erase failure, or other causes.

[0047] In operation **430**, the controller may replace bad MLC blocks in the Flash memory with spare blocks from the MLC spare block pool.

[0048] In operation **440**, the controller may determine if the number of blocks in the SLC spare block pool is less than a minimum SLC spare pool threshold (e.g., the minimum number of SLC blocks necessary for smooth function of the SLC memory cells and/or the Flash memory device). If the number of blocks in the SLC spare block pool is less than a minimum SLC spare pool threshold, a process or the controller may proceed to operation **450**. If the number of blocks in the SLC spare block pool is equal to or greater than a minimum SLC spare pool threshold, a process or the controller may proceed to operation **420**.

[0049] In operation **450**, the controller may re-allocate spare blocks from the MLC spare block pool to the SLC spare block pool. The controller may, in some embodiments, transfer or re-allocate MLC spare blocks whose P/E cycles have exceeded the maximum P/E cycle rating for an MLC block but not the maximum P/E cycle rating for an SLC block. Any MLC block, or other type of block, that has not exceeded the maximum P/E cycle rating for SLC blocks may be re-allocated to the SLC spare block pool.

[0050] In operation **460**, the controller may determine if the number of blocks in the MLC spare block pool is less than a minimum MLC spare pool threshold. If the number of blocks in the MLC spare block pool is less than a minimum MLC spare pool threshold, a process or controller may proceed to operation **470**. If the number of blocks in the MLC spare block pool is greater than or equal to a minimum MLC spare pool threshold, a process or controller may proceed to operation **420**.

[0051] In operation **470**, the controller may declare an end of life of the Flash memory array.

[0052] Other operations or orders of operations may be used.

[0053] Embodiments of the invention may demonstrate one or move advantages, including, without requirement or limitation, ensuring optimal usage of all blocks throughout the life of the Flash memory device, increased spare block resources for the buffer and cache processes within the Flash device, longer Flash memory device lifespan, and/or reducing environmental waste from expired flash memory devices.

[0054] While certain features of the invention have been illustrated and described herein, many modifications, substitutions, changes, and equivalents may occur to those skilled in the art. It is, therefore, to be understood that the appended

claims are intended to cover all such modifications and changes as fall within the true spirit of the invention.

What is claimed is:

1. A method of operating a Flash memory array comprising:

allocating a first portion of spare blocks in the Flash memory as a single level cell (SLC) spare block pool, and a second portion of spare blocks in the Flash memory as a multi-level cell (MLC) spare block pool;

replacing bad SLC blocks in the Flash memory with spare blocks from the SLC spare block pool; and

if the number of spare blocks in the SLC spare block pool is less than a minimum SLC spare pool threshold, then re-allocating spare blocks from the MLC spare block pool to the SLC spare block pool.

2. The method of claim **1**, further comprising:

re-allocating MLC blocks reaching a maximum MLC bearable cycle count to the SLC spare block pool.

3. The method of claim **2**, wherein the maximum MLC bearable cycle count comprises a predetermined number of P/E cycles.

4. The method of claim **1**, further comprising:

declaring an end of life of the Flash memory array when the number of blocks in the MLC spare block pool is less than a minimum MLC spare pool threshold.

5. The method of claim **1**, wherein the minimum SLC spare pool threshold comprises a minimum number of spare blocks required for smooth operation.

6. A Flash memory device comprising:

a plurality of Flash memory blocks; and

a controller to:

allocate a first portion of spare blocks in the Flash memory as a single level cell (SLC) spare block pool, and a second portion of spare blocks in the Flash memory as a multi-level cell (MLC) spare block pool;

replace bad SLC blocks in the Flash memory with spare blocks from the SLC spare block pool; and

if the number of spare blocks in the SLC spare block pool is less than a minimum SLC spare pool threshold, then re-allocate spare blocks from the MLC spare block pool to the SLC spare block pool.

7. The device of claim **6**, wherein the controller is further configured to:

re-allocate MLC blocks reaching a maximum MLC bearable cycle count to the SLC spare block pool.

8. The device of claim **7**, wherein the maximum MLC bearable cycle count is a predetermined number of P/E cycles.

9. The device of claim **6**, wherein the controller is further configured to:

declare an end of life of the Flash memory array when the number of blocks in the MLC spare block pool is less than a minimum MLC spare pool threshold.

10. The device of claim **6**, wherein the minimum SLC spare pool threshold comprises a minimum number of spare blocks required for smooth operation.

11. A method of operating a non-volatile memory array comprising:

allocating N number of portions of spare blocks in the non-volatile memory as N number of multi-level cell (MLCx(N)) spare block pools;

exchanging bad MLCx(j) blocks in the non-volatile memory with spare blocks from the MLCx(j) spare block pool; and

if the number of spare blocks in an MLCx(j) spare block pool is less than a minimum MLCx(j) spare pool threshold, then transferring spare blocks from the MLCx(j+1) spare block pool to the MLCx(j) spare block pool.

12. The method of claim **11**, wherein an MLCx1 block comprises an SLC spare block.

13. The method of claim **11**, wherein the non-volatile memory array comprises a Flash memory array.

14. The method of claim **11**, wherein N is greater or equal to three.

15. The method of claim **11**, further comprising:

transferring MLCx(j+1) blocks reaching a maximum MLCx(j+1) bearable cycle count to the MLCx(j) spare block pool.

16. The method of claim **11**, further comprising:

declaring an end of life of the non-volatile memory array when the number of blocks in the MLCxN spare block pool is less than a minimum MLCxN spare pool threshold.

* * * * *