CLAIMS

1. A method for transaction log recovery, comprising:

examining a number of entries saved in a transaction log to determine a write pattern;

reading the memory based on the write pattern;

updating the transaction log with information associated with data read from the memory based on the write pattern; and

updating a logical address (LA) table using the transaction log.

2. The method of claim 1, wherein reading the memory based on the write pattern includes reading pages in memory.

3. The method of any one of claims 1-2, wherein the transaction log includes information about the location where data had most recently been written in the memory.

4. The method of claim 2, wherein the transaction log did not have information associated with the pages read prior to being updated.

5. The method of claim 2, wherein updating the transaction log comprises updating the transaction log with information associated with pages read in memory that include data.

6. The method of any one of claims 1-2, wherein reading the memory comprises reading pages in the memory in the order based on the write patternuntil an erased page is found.

7. The method of claim 2, further including locating a partially written block by reading the first and last pages of a number of blocks and determining the type of data included in the partially written block when the pages read in the memory in the order based on the write pattern are in a block that does not include erased pages.

8. The method of claim 7, further including reading pages of the partially written block that includes host data.

9. The method of claim 8, further including updating the transaction log with information associated with pages read in the partially written block that contains data.

10. A method for operating a memory system, comprising:

reading pages in the memory system that do not have associated data in a transaction log in an order based on a write pattern;

updating a transaction log with information associated with the pages read based on the write pattern; and

updating a logical address (LA) table in non-volatile memory using the updated transaction log.

11. The method of claim 10, further comprising, prior to a power interrupt, storing the LA table in the non-volatile memory on a periodic basis, the stored LA table being a copy of a LA table in volatile memory.

12. The method of claim 11, wherein the LA table in the volatile memory is updated after each write operation in the memory system.

13. The method of claim 11, wherein, prior to updating the LA table, the LA table in non-volatile memory does not include information about write operations that occurred between a last time the LA table was stored in non-volatile memory and a power interruption.

14. The method of any one of claims 10-11, further comprising copying the updated LA table to volatile memory upon power up of the memory system after a power interruption.

15. The method of any one of claims 10-11, wherein the write pattern is based on a write pattern algorithm that determines where to write data in the memory system.

16. The method of claim 15, wherein the write pattern includes information about the location where data would have been written next in the memory system.

17. The method of claim 16, wherein the method includes verifying a valid write occurred at the page read based on the write pattern.

18. A method for transaction log recovery in a memory system, comprising:

reading pages in an order based on a write pattern to find write operation information that was not in a transaction log or a logical address (LA) table in non-volatile memory;

updating the transaction log using the found write operation information;

updating the LA table in non-volatile memory using the updated transaction log; and

storing the LA table in volatile memory.

19. The method of claim 18, wherein finding information about the write operation includes determining the location where data would have been written next in the memory system from the write pattern.

20. The method of claim 19, wherein determining the location where the data would have been written next comprises using a write pattern algorithm.

21. The method of any one of claims 18-19, wherein updating the transaction log comprises updating a last page of the transaction log that became corrupt after the power interrupt.

22. The method of any one of claims 18-19, wherein the method includes storing a copy of the LA table in the volatile memory on a periodic basis.

23. The method of any one of claims 18-19, wherein the method includes recording information about read and write operations in the transaction log for operations that occur after storing a copy of the LA table in the non-volatile memory.

24. The method of any one of claims 18-19, wherein updating the LA table in the non-volatile memory using the updated transaction log comprises adding information about a write that occurred between a last time the LA table was copied into the non-volatile memory and a power interrupt.

25. A memory system, comprising:

non-volatile memory configured to store a logical address (LA) table and a transaction log; and

a controller configured to:

determine a write pattern by examining information in a transaction log that indicates where data was written in memory;

read pages in the memory in an order based on the write pattern;

update the transaction log with information associated with the pages read in memory; and

update the logical address (LA) table using the transaction log.

26. The memory system of claim 25, wherein the transaction log is configured to record information about writes occurring in the memory system after storing the LA table in non-volatile memory.

27. The memory system of claim 25, wherein the controller is configured to recreate the last page of transaction log information in a transaction log by reading pages in memory in the order based on the write pattern.

28. The memory system of any one of claims 25-27, wherein the controller is configured to create the write pattern based on a write pattern algorithm.

What is Claimed is:

1. A method for transaction log recovery, comprising:

examining a number of entries saved in a transaction log to determine a write pattern;

reading the memory based on the write pattern;

updating the transaction log with information associated with data read from the memory based on the write pattern; and

updating a logical address (LA) table using the transaction log.

2. The method of claim 1, wherein reading the memory based on the write pattern includes reading pages in memory.

3. The method of claim 1, wherein the transaction log includes information about the location where data had most recently been written in the memory.

4. The method of claim 2, wherein the transaction log did not have information associated with the pages read prior to being updated.

5. The method of claim 2, wherein updating the transaction log comprises updating the transaction log with information associated with pages read in memory that include data.

6. The method of claim 1, wherein reading the memory comprises reading pages in the memory in the order based on the write pattern until an erased page is found.

7. The method of claim 2, further including locating a partially written block by reading the first and last pages of a number of blocks and determining the type of data included in the partially written block when the pages read in the memory in the order based on the write pattern are in a block that does not include erased pages.

8. The method of claim 7, further including reading pages of the partially written block that includes host data.

9. The method of claim 8, further including updating the transaction log with information associated with pages read in the partially written block that contains data.

10. A method for operating a memory system, comprising:

reading pages in the memory system that do not have associated data in a transaction log in an order based on a write pattern;

updating a transaction log with information associated with the pages read based on the write pattern; and

updating a logical address (LA) table in non-volatile memory using the updated transaction log.

11. The method of claim 10, further comprising, prior to a power interrupt, storing the LA table in the non-volatile memory on a periodic basis, the stored LA table being a copy of a LA table in volatile memory.

12. The method of claim 11, wherein the LA table in the volatile memory is updated after each write operation in the memory system.

13. The method of claim 11, wherein, prior to updating the LA table, the LA table in non-volatile memory does not include information about write operations that occurred between a last time the LA table was stored in non-volatile memory and a power interruption.

14. The method of claim 10, further comprising copying the updated LA table to volatile memory upon power up of the memory system after a power interruption.

15. The method of claim 10, wherein the write pattern is based on a write pattern algorithm that determines where to write data in the memory system.

16. The method of claim 15, wherein the write pattern includes information about the location where data would have been written next in the memory system.

17. The method of claim 16, wherein the method includes verifying a valid write occurred at the page read based on the write pattern.

18. A method for transaction log recovery in a memory system, comprising:

reading pages in an order based on a write pattern to find write operation information that was not in a transaction log or a logical address (LA) table in non-volatile memory ;

updating the transaction log using the found write operation information;

updating the LA table in non-volatile memory using the updated transaction log; and

storing the LA table in volatile memory.

19. The method of claim 18, wherein finding information about the write operation includes determining the location where data would have been written next in the memory system from the write pattern.

20. The method of claim 19, wherein determining the location where the data would have been written next comprises using a write pattern algorithm.

21. The method of claim 18, wherein updating the transaction log comprises updating a last page of the transaction log that became corrupt after the power interrupt.

22. The method of claim 18, wherein the method includes storing a copy of the LA table in the volatile memory on a periodic basis.

23. The method of claim 18, wherein the method includes recording information about read and write operations in the transaction log for operations that occur after storing a copy of the LA table in the non-volatile memory.

24. The method of claim 18, wherein updating the LA table in the non-volatile memory using the updated transaction log comprises adding information about a write that occurred between a last time the LA table was copied into the non-volatile memory and a power interrupt.

25. A memory system, comprising:

non-volatile memory configured to store a logical address (LA) table and a transaction log; and

a controller configured to:

determine a write pattern by examining information in a transaction log that indicates where data was written in memory;

read pages in the memory in an order based on the write pattern;

update the transaction log with information associated with the pages read in memory; and

update the logical address (LA) table using the transaction log.

26. The memory system of claim 25, wherein the transaction log is configured to record information about writes occurring in the memory system after storing the LA table in non-volatile memory.

27. The memory system of claim 25, wherein the controller is configured to recreate the last page of transaction log information in a transaction log by reading pages in memory in the order based on the write pattern.

28. The memory system of claim 25, wherein the controller is configured to create the write pattern based on a write pattern algorithm.

29. A memory system, comprising:

non-volatile memory, wherein the non-volatile memory is configured to store a logical address (LA) table and a transaction log; and

a controller configured to:

update the transaction log by reading pages in an order based a write pattern, and

update the LA table with information missing from the LA table after a power interruption using the updated transaction log.

30. The memory system of claim 29, wherein the controller is further configured to store a copy of the updated LA table in volatile memory.

31. The memory system of claim 30, wherein the controller is configured to continually update the updated LA table in the volatile memory and store a copy of the continually updated LA table in the non-volatile memory at least once every 300 seconds.

32. The memory system of claim 29, wherein the controller is configured to determine the write pattern using the transaction log.

33. The memory system of claim 32, wherein the controller is configured to determine where data would have been written next in the memory system using the write pattern.

34. The memory system of claim 29, wherein the LA table is a logical block address (LBA) table.