PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06F 12/02 | A1

(11) International Publication Number: WO 94/23369

(43) International Publication Date: 13 October 1994 (13.10.94)

(21) International Application Number: PCT/US94/03168

(22) International Filing Date: 23 March 1994 (23.03.94)

(30) Priority Data:
08/037,893 26 March 1993 (26.03.93) US

(71) Applicant: CIRRUS LOGIC, INC. [US/US]; 3100 Western Warren Avenue, Fremont, CA 94538 (US).

(72) Inventors: ASSAR, Mahmud; 14525 Shadowlane Court, Morgan Hill, CA 95037 (US). NEMAZIE, Siamack; 1253 Quai Creek Circle, San Jose, CA 95120 (US). ESTAKHRI, Petro; 7966 Foothill Knolls, Pleasanton, CA 94566 (US).

(74) Agents: HAVERSTOCK, Thomas, B. et al.; Haverstock, Medlen & Carroll, Suite 2200, 220 Montgomery Street, San Francisco, CA 94104 (US).

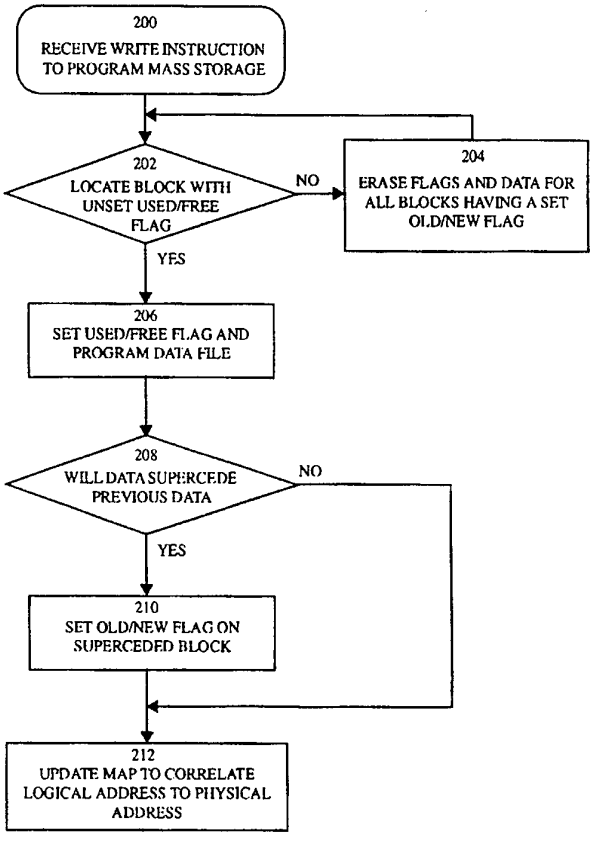
(81) Designated States: CA, JP, KR, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: FLASH MEMORY MASS STORAGE ARCHITECTURE

(57) Abstract

A semiconductor mass storage system (100) and architecture can be substituted for a rotating hard disk. The system and architecture avoid an eras cycle each time information stored in the mass storage is changed. Erase cycles are avoided by programming an altered data file into an empty mass storage block (step 202-206) rather than over itself as hard disk would. Periodically, the mass storage will need to be cleaned up (step 204). Secondly, means are provied for evenly using all blocks in the mass storage (Fig. 7). These advantages are achieved through the use of several flags (200), a map to correlate a logical address of a block to a physical address of that block (308, 408) and a count register for each block. In particular, flags are provided for defective blocks (118), used blocks (112), old version of a block (104, 116), a count to determine the number of times a block has been erased and written and erase inhibit flag (200).



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	**GB**	United Kingdom	**MR**	Mauritania
AU	Australia	**GE**	Georgia	**MW**	Malawi
BB	Barbados	**GN**	Guinea	**NE**	Niger
BE	Belgium	**GR**	Greece	**NL**	Netherlands
BF	Burkina Faso	**HU**	Hungary	**NO**	Norway
BG	Bulgaria	**IE**	Ireland	**NZ**	New Zealand
BJ	Benin	**IT**	Italy	**PL**	Poland
BR	Brazil	**JP**	Japan	**PT**	Portugal
BY	Belarus	**KE**	Kenya	**RO**	Romania
CA	Canada	**KG**	Kyrgystan	**RU**	Russian Federation
CF	Central African Republic	**KP**	Democratic People's Republic of Korea	**SD**	Sudan
CG	Congo			**SE**	Sweden
CH	Switzerland	**KR**	Republic of Korea	**SI**	Slovenia
CI	Côte d'Ivoire	**KZ**	Kazakhstan	**SK**	Slovakia
CM	Cameroon	**LI**	Liechtenstein	**SN**	Senegal
CN	China	**LK**	Sri Lanka	**TD**	Chad
CS	Czechoslovakia	**LU**	Luxembourg	**TG**	Togo
CZ	Czech Republic	**LV**	Latvia	**TJ**	Tajikistan
DE	Germany	**MC**	Monaco	**TT**	Trinidad and Tobago
DK	Denmark	**MD**	Republic of Moldova	**UA**	Ukraine
ES	Spain	**MG**	Madagascar	**US**	United States of America
FI	Finland	**ML**	Mali	**UZ**	Uzbekistan
FR	France	**MN**	Mongolia	**VN**	Viet Nam
GA	Gabon				

FLASH MEMORY MASS STORAGE ARCHITECTURE

Field of the Invention

This invention relates to the field of mass storage for computers. More particularly, this invention relates to an architecture for replacing a hard disk with a semiconductor non-volatile memory and in particular flash memory.

Background of the Invention

Computers have used rotating magnetic media for mass storage of data, programs and information. Though widely used and commonly accepted, such hard disk drives suffer from a variety of deficiencies. Because of the rotation of the disk, there is an inherent latency in extracting information from a hard disk drive.

Other problems are especially dramatic in portable computers. In particular, hard disks are unable to withstand many of the kinds of physical shock that a portable computer will likely sustain. Further, the motor for rotating the disk consumes significant amounts of power decreasing the battery life for portable computers.

Solid state memory is an ideal choice for replacing a hard disk drive for mass storage because it can resolve the problems cited above. Potential solutions have been proposed for replacing a hard disk drive with a semiconductor memory. For such a system to be truly useful, the memory must be non-volatile and alterable. The inventors have determined that flash memory is preferred for such a replacement. It should be noted that E^2 PROM is also suitable as a replacement for a hard disk drive.

Flash memory is a single transistor memory cell which is programmable through hot electron injection and erasable through Fowler-Nordheim tunneling. The programming and erasing of such a memory cell requires current to pass

through the dielectric surrounding a floating gate electrode. Because of this, such types of memory have a finite number of erase-write cycles. Eventually, the dielectric will fail. Manufacturers of flash cell devices specify the limit for the number erase-write cycles as between 10,000 and 100,000. Accordingly, unlike rotating magnetic media, a flash memory mass storage device does not have an indefinite lifetime.

Another requirement for a semiconductor mass storage device to be successful is that its use in lieu of a rotating media hard disk mass storage device be transparent to the system designer and the user. In other words, the designer of a computer incorporating such a semiconductor mass storage device could simply remove the hard disk and replace it with a semiconductor mass storage. All presently available commercial software should operate on a system employing such a semiconductor hard disk without the necessity of any modification.

SunDisk proposed an architecture for a semiconductor mass storage using flash memory at the Silicon Valley PC Design Conference July 9, 1991. That mass storage system included read-write block sizes of 512 Bytes (or multiples thereof) just like IBM PC compatible hard disk sector sizes. (IBM PC is a trademark of IBM Corporation.) During an erase cycle, an entire block is first fully programmed and then erased.

As in conventional hard disks, it appears in the SunDisk architecture that there is an erase-before-write cycle each time data is changed in the mass storage. Thus, if a program or data block is to be changed, the data is written to RAM and appropriately changed, the flash block is fully programmed, then erased and then reprogrammed to the new memory condition. Unlike a hard disk device, in a flash memory device an erase cycle is slow which can significantly

reduce the performance of a system utilizing flash memory as its mass storage.

Though such an architecture provides a workable semiconductor mass storage, there are several inefficiencies. First of all, each time a memory block is changed, there is a delay to the entire system due to the necessary erase-before-write cycle before reprogramming the altered information back into the block. The overhead associated with erase-before-write cycles is costly in terms of system performance.

Secondly, hard disk users typically store both information which is rarely changed and information which is frequently changed. For example, a commercial spread sheet or word processing software programs stored on a user's system are rarely, if ever, changed. However, the spread sheet data files or word processing documents are frequently changed. Thus, different sectors of a hard disk typically have dramatically different usage in terms of the number of times the information stored thereon is changed. While this disparity has no impact on a hard disk because of its insensitivity to data changes, in a flash memory device, this variance can cause sections of the mass storage to wear out and be unusable significantly sooner than other sections of the mass storage.

Summary of the Invention

The present invention discloses two primary algorithms and an associated hardware architecture for a semiconductor mass storage device. It will be understood that data file in this patent document refers to any computer file including commercial software, a user program, word processing software document, spread sheet file and the like. The first algorithm provides means for avoiding an erase-before-write cycle when writing a modified data file

back onto the mass storage device. Instead, no erase is performed and the modified data file is written onto an empty portion of the mass storage. In addition, the second algorithm prevents any portion of the mass storage from being erased a substantially larger number of times than any other portion. This prevents any one block of the mass storage from failing and becoming unusable earlier than any other block thereby extending the life of the entire mass storage.

The semiconductor mass storage architecture has blocks sized to conform with commercial hard disk sector sizes. The blocks are individually erasable. In one embodiment, the semiconductor mass storage of the present invention can be substituted for a rotating hard disk with no impact to the user, so that such a substitution will be transparent. Means are provided for avoiding the erase-before-write cycle each time information stored in the mass storage is changed. (The erase cycle is understood to include, fully programming each bit in the block to be erased, and then erasing all the bits in the block.)

According to the first algorithm, erase cycles are avoided by programming an altered data file into an empty mass storage block rather than over itself after an erase cycle of that block as done on a conventional hard disk. This would ordinarily not be possible when using conventional mass storage because the central processor and commercial software available in conventional computer systems are not configured to track continually changing physical locations of data files. The present invention includes a programmable map to maintain a correlation between the logical address 308 and the physical address 408 of the updated information files.

Periodically, the mass storage will fill up because there have been no erase cycles. At such times, the mass

storage needs to be cleaned up with a multi-sector erase as fully described in the detailed description below.

According to the second algorithm, means are provided for evenly using all blocks in the mass storage. A counter tracks the number of times each block is erased. A programmable maximum value for the counter is also provided. As the number of erase cycles for a block becomes one less than the maximum, the block is erased one last time and written with another file having a then smallest number of erase cycles. It is also prevented from being erased thereafter by setting its erase inhibit flag. After all blocks approach this maximum, all the erase counters and inhibit flags are cleared and the second algorithm is then repeated. In this way, no block can be erased a substantial number of times more than any other block.

These advantages are achieved through the use of several flags and a count register for each block. In particular, flags are provided for defective blocks, used blocks, old version of a block, a count to determine the number of times a block has been erased and written and an erase inhibit flag.

Brief Description of the Drawings

Figure 1 shows an architecture for a semiconductor mass storage.

Figure 2 shows the architecture of Figure 1 wherein the data in one block has been altered and stored in a new physical address.

Figure 3 shows a block diagram of an erase cycle usage according to algorithm 1 of the present invention.

Figure 4 shows a simplified block diagram of the old/new flag system integrally formed with the memory.

Figure 5 shows a flow chart block diagram for

algorithm 1.

Figure 6 shows an additional architecture according to the preferred embodiment of the present invention.

Figure 7 shows a flow chart block diagram of algorithm 2 of the present invention.

Figure 8 shows a flow chart block diagram of a read algorithm according to the present invention.

Detailed Description of the Preferred Embodiment

Figure 1 shows an architecture for a semiconductor mass storage according to the present invention. In the preferred embodiment, all of the memory storage is flash EEPROM. It is possible to substitute E^2PROM for some or all of the data bits shown. A memory storage 100 is arranged into N blocks of data from zero through N-1. Each of the blocks of data is M Bytes long. In the preferred embodiment, each block is 512 Bytes long to correspond with a sector length in a commercially available hard disk drive. In addition to the memory data block 102, a flag 104 is directly related to each data block 102. The memory 100 can contain as much memory storage as a user desires. An example of a mass storage device might include 100 MByte of addressable storage.

A non-volatile content addressable memory (CAM) 106 is associated with the memory storage 100. In the preferred embodiment, the CAM 106 is formed of flash memory. The CAM 106 can also be E^2PROM. There is one entry in the CAM 106 for every one of the N blocks in the mass storage 100. Each entry includes a number of fields which will be described below. The CAM 106 is also formed of a non-volatile memory because loss of its information would make retrieval of the data files stored in the mass storage 100 impossible.

As described above in the Background of the Invention, conventional computer systems are not configured to track

continually changing physical locations of data files. According to the present invention, each time a data file is changed it is stored into a new physical location in the mass storage. Thus, implementation of the architecture of the present invention requires a mapping of the logical address 308, i.e., the address where the computer system believes the data file is stored to the physical address 408, i.e., the actual location the data file can be found is stored in the mass storage.

The logical address 308 portion of the map 108 and the flags 112, 116 and 118 form part of the CAM 106. It is possible to use other storage means than a CAM to store the address map, such as a look-up table. However, a CAM is the most efficient means known to the inventors. It is not necessary that the physical address 408 portion of the map 108 form part of the CAM. Indeed, the physical address 408 portion of the map 108 can be ordinary flash memory, E^2PROM or even ROM. If ROM is selected for the physical address 408 array of the map 108, a defect in the ROM will prevent the block corresponding to that physical address 408 from ever being addressed. Accordingly, a changeable nonvolatile memory is preferred. Note that any replacement circuit for the CAM should be nonvolatile. Otherwise, loss or removal of power to the system will result in loss of the ability to find the data files in the mass storage.

Assume for example that a user is preparing a word processing document and instructs the computer to save the document. The document will be stored in the mass storage system as shown in Figure 1. The computer system will assign it a logical address 308, for example 526H. The mass storage system of the present invention will select a physical address 408 of an unused block or blocks in the mass storage 100 for storing the document, e.g. 728H. That map correlating the logical address 308 to the physical

address 408 is stored in the CAM 106. As the data is programmed, the system of the present invention also sets the used/free flag 112 to indicate that this block has been written without being erased. The used/free flag 112 also forms a portion of the CAM 106. One used/free flag 112 is provided for each entry of the CAM 106.

Later, assume the user retrieves the document, makes a change and again instructs the computer to store the document. To avoid an erase-before-write cycle, the system of the present invention provides means for locating a block having its used/free flag 112 unset (not programmed) which indicates that the associated block is erased. The system then sets the used/free flag for the new block 114 (Figure 2) and then stores the modified document in that new block 114. Next, the system sets the old/new flag 116 of the previous version of the document indicating that this is an old unneeded version of the document. Lastly, the system updates the correlation between the logical address 308 and the actual physical address 408. In this way, the system of the present invention avoids the overhead of an erase cycle which is required in the erase-before-write of conventional systems to store a modified version of a previous document.

The writing to mass storage process outlined above is repeated until the entire mass storage memory 100 has been filled. A full mass storage is indicated by no unset used/free flags 112 in the CAM 106. At that time a multi-sector erase is necessary and those blocks in the memory 100 and their associated CAM 106 entries having an old/new flag 116 set are all erased simultaneously. Note that it is not necessary for 100% of the blocks to have a set used/free flag 112 for a multi-sector erase to occur. For example, if a data file requiring three blocks were being written and only two blocks having unset used/free flags 112 were available a multi-sector erase can be run.

A simultaneous erase is not needed with prior art implementations because those embodiments utilize an erase-before-write cycle rather than retaining superseded versions of data files. In such circuits a latch of volatile logic circuits is set to couple the voltage necessary to erase the flash cells in the block. Because of the likely large number of memory blocks in the mass storage 100, if the CAM 106 and mass storage 100 are on the same integrated circuit (chip) coupling the old/new flag 116 to the latches in parallel would typically be very expensive in terms of surface area of the chip and coupling the old/new flags 116 serially to the latches would be expensive in terms of system performance. If the CAM 106 and the mass storage 100 are on separate chips, it is doubtful that either device could have sufficient I/O capability to interconnect the old/new flags 116 to the latches in parallel and thus, the system would suffer from a serial transfer of that information for a multi-sector erase.

Because of these problems it is preferable that no updating of the latches be performed prior to an erase of all blocks having a set old/new flag 116. To avoid this step, a plurality of old/new flag systems 104 are intimately associated with each block in the memory 102 and is programmed by the same sequence of instructions as the old/new flag 116 of the CAM 106.

Figure 4 shows a simplified block diagram of the old/new flag system 104 which includes a non-volatile bit 120 having data which mirrors the old/new flag 116. In addition there is a volatile latch 122 coupled to receive the data in the bit 120 from the latch during an erase cycle. At the time of an erase, the data in each of the bits 120 is simultaneously coupled to each appropriate ones of the latches 122 under control of a load signal coupled to each latch 122 over a load line L. Upon receiving a signal

to perform the erase, the latch for every block having its associated bit 120 set then couples the voltage necessary to perform an erase of that block and its associated bit 120. After the erase is complete and verified, all the latches 122 are individually reset to a predetermined state under control of a reset signal coupled to each latch 122 over a reset line R.

For certain applications of the present invention, especially for low power portable computers, a simultaneous erase of all blocks having their respective old/new flags set may be undesirable. For such applications, the blocks can be segregated into groups of blocks. Each group has a unique control line to load the latches from the nonvolatile bits. In this mode, during an erase cycle, the control lines are sequentially activated and the groups of blocks sequentially erased.

Figure 5 shows algorithm 1 according to the present invention. When the system of the present invention receives an instruction to program data into the mass storage (step 200), then the system attempts to locate a free block (step 202), i.e., a block having an unset (not programmed) used/free flag. If successful, the system sets the used/free flag for that block and programs the data into that block (step 206).

If on the other hand, the system is unable to locate a block having an unset used/free flag, the system erases the flags (used/free and old/new) and data for all blocks having a set old/new flag (step 204) and then searches for a block having an unset used/free flag (step 202). Such a block has just been formed by step 204. The system then sets the used/free flag for that block and programs the data file into that block (step 206).

If the data file is a modified version of a previously existing file, the system must prevent the superseded

version from being accessed. The system determines whether the data file supersedes a previous data file (step 208). If so, the system sets the old/new flag associated with the superseded block (step 210). If on the other hand, the data file to be stored is a newly created data file, the step of setting the old/new flag (step 210) is skipped because there is no superseded block. Lastly, the map for correlating the logical address 308 to the physical address 408 is updated (step 212).

By following the procedure outlined above, the overhead associated with an erase cycle is avoided for each write to the memory 100 except for periodically. This vastly improves the performance of the overall computer system employing the architecture of the present invention.

In the preferred embodiment of the present invention, the programming of the flash memory follows the procedure commonly understood by those of ordinary skill in the art. In other words, the program impulses are appropriately applied to the bits to be programmed and then compared to the data being programmed to ensure that proper programming has occurred. In the event that a bit fails to be erased or programmed properly, a defect flag 118 in the CAM 106 is set preventing that block from being used again.

In addition to saving the overhead of the erase cycle all but periodically, utilization of the present invention tends to more evenly distribute the erase cycles amongst certain portions of the blocks of the mass storage. Figure 3 schematically shows the types of information stored in utilizing a mass storage media 150. One portion of the mass storage 150 contains commercial applications software 152 such as word processing, spreadsheet, calendaring, calculators and the like. These portions of the mass storage 150 rarely, if ever, require an erase-reprogram cycle according to the algorithm described above.

A second section of the mass storage 150 contains user data 154. The user data 154 is frequently altered requiring the information to be reprogrammed into blocks of the free space 156 under the algorithm described above. A third portion of the mass storage 150 contains free space 156 of unprogrammed blocks.

By following the algorithm above, the storage blocks in the portions 154 and 156 of the memory 150 will recycle data files and thus be erased and reprogrammed significantly more often than the commercial applications software portion 152 of the memory 150. Accordingly, the mass storage 150 will wear out more quickly in the user data 154 and the free space 156 sections of the memory requiring earlier replacement than in sections 152 of the mass storage having data files which are rarely changed. As the number of free blocks diminishes providing a smaller number of blocks through which to recycle data files, the remaining blocks become erased more frequently exacerbating the problem.

A second algorithm is provided for leveling erase cycles amongst all the blocks within the entire mass storage device as shown in Figure 6. A counter is provided for each block to count the number of times each block has been erased and reprogrammed. An erase inhibit flag is also provided for each block. Once the erase count has reached the maximum for any block, the erase inhibit flag is set for that block. After that time that block cannot be erased until a clean-out erase is performed. Referring to Figure 3, if only algorithm 1 is used eventually all of the blocks in the user data 154 and the free space 156 portions of the mass storage 150 will reach the maximum count and have their respective erase inhibit flags set. Because of this, a reallocation of the rarely erased data files stored in the memory 152 is made into the memory 154 and/or 156. In this way, sections of the mass storage which have been

erased numerous times are programmed with a reallocated data file which is rarely changed thereby allowing all sections of the mass storage to eventually approach parity of erase cycles. Like the multi-sector erase, a clean-out erase can be performed in the event that there is insufficient available storage for a data file presently being performed. For example, if all but two blocks have their respective erase inhibit flags set, and a three or more block data file is being programmed, a clean-out erase can be performed to provide sufficient storage for the data file.

Once the erase inhibit flag is set for all the blocks, indicating that all the blocks have achieved parity in erase cycles, the erase inhibit and erase count registers are erased and the cycle is repeated. The selection of the maximum count depends upon the system requirements. As the value for the maximum count increases, the disparity between erase count cycles of various blocks can also increase. However, because data is shifted as a result of achieving maximum erase count this process of smoothing cycles throughout the mass storage of itself introduces additional erase cycles because a block of information is transferred from a physical block having few erases to a block having the maximum number of erases. Accordingly, though low maximum count values reduce the disparity between erase cycles amongst the blocks it also increases the number of erase cycles to which the blocks are subjected. Accordingly, individual users may select an erase count depending upon the system needs.

In the preferred embodiment, algorithm 2 is merged with algorithm 1 as shown in Figure 7. An instruction is provided by the computer system to write a data file to the mass storage (step 230) which starts the combined algorithm 1 and algorithm 2 sequence. It is first determined whether the mass storage is full (step 232). If the mass storage is

not full, i.e., it has a block with its used/free flag unset, the algorithm continues and stores the data file into such a block (step 234).

If on the other hand, it is determined that there are no free blocks, then it is next determined whether there are any blocks which have both the old/new flag set AND the erase inhibit flag unset (step 236). If there are no blocks which have both the old/new flag set AND the erase inhibit flag unset (step 236), the system of the present invention erases the data file, used/free flag and old/new flag in each block having its old/new flag set, and erases the counter and erase inhibit flag for every block (step 238). Step 238 is also performed in the event there are insufficient blocks remaining to store a pending data file. The algorithm then returns to block (step 232) to determine whether the disk is full.

If the system can find a block having both the old/new flag set AND the erase inhibit flag unset (step 236), then the system executes an erase procedure and erases the data file, used/free flag and old/new flag in each block having its old/new flag set. The counter is incremented and the erase inhibit flag for such blocks is not disturbed.

It is then determined whether any block having its used/free flag unset has its counter at the maximum count (step 242). If not, then the system of the present invention returns to decision step 232 and investigates again whether there is any block having its used/free flag unset (step 232).

On the other hand, if there is a block having its erase count at the maximum value, a data file is copied from another block having the then least count value (step 244) into the location having COUNT = $COUNT_{Max}$. The erase inhibit flag is then set (step 244). Note that a data file will not be copied from a block having its erase count at one less

than the maximum value, $COUNT_{Max}$-1. Making such a reallocation from a source block having $COUNT_{Max}$-1 to a destination block having $COUNT_{Max}$ results in having both blocks at $COUNT_{Max}$ and no net gain. Further, the block previously having its erase count at $COUNT_{Max}$-1 is erased to no advantage, thus the erase cycle for that block would be wasted.

The old/new flag from the source block is then set (step 246) so that it can be erased during the next execution of an erase step 240. In that way the source block can be used for storage until its erase count reaches maximum and its erase inhibit flag is set. The algorithm then returns to step 242 to determine whether there are now any blocks having an unset used/free flag with an erase count less than $COUNT_{Max}$. It will be understood that each time a data file is programmed or moved according to the algorithm of Figure 7 that the map in the CAM which correlates the logical address 308 to physical address 408 is updated so that the computer system can always access the data files.

The efficiency of these algorithms has been tested by simulation. In the simulation it was assumed that the mass storage was 50% filled with data files that are not changed, 30% with data files that are routinely changed and 20% empty. Of the 30% of the data files that are routinely changed, ⅛ are rewritten 70% of the time, ⅛ are rewritten 25% of the time and ⅛ are rewritten 5% of the time. The simulation showed that the algorithm 1 improves the number of cycles until any block has reached failure by between six and seven times and algorithm 2 by approximately two times over the improvement gained using algorithm 1 alone. Depending upon the design criterion of a target system, it is possible to utilize either algorithm 1, algorithm 2 or the preferred merged algorithm. Algorithm 1 and the merged

algorithm have been described above.

In the preferred embodiment, a bit is programmed into the counter for each erase cycle rather than using binary counting. Thus, an eight bit counter register would only be able to count to eight. This avoids having to erase the counter and then reprogramming it with an incremented value as would be necessary for binary counting. This is preferred because it avoids having to temporarily store the count value for all of the blocks being erased. By programming a bit for each, the counter registers need not be erased until all the blocks reach maximum count and there is a general erase.

Because the mass storage apparatus of the present invention can accommodate large data storage, it is likely that many blocks will be flagged for a clean-out erase. Either a temporary volatile storage would be necessary for each block to store the previous count value prior to incrementing and reprogramming or the erase and updating of the counters would have to be done one after the other. One solution requires integrated circuit surface area and the other degrades performance. Note however, that if binary counting is desired the erase counter can be erased each time the block is erased and immediately reprogrammed. Because this will happen only during the periodic erase cycle described relative to the first algorithm some system designers may find this acceptable.

The read algorithm according to the present invention is shown in Figure 8. A read instruction is received by the mass storage apparatus of the present invention from the computer system (step 270). Concurrent with receiving the read instruction, the system also receives the logical address 308 of the data file needed by the computer system (step 271). The apparatus of the present invention concatenates all the appropriate flags to the logical

address 308 including having a set used/free flag, and unset new/old and defect flags (step 272). If a match is found in the CAM (step 273), the data file is read (step 275) otherwise a signal is returned to the computer system that the data file was not found (step 274).

The present invention is described relative to a preferred embodiment. Modifications or improvements which become apparent to one of ordinary skill in the art after reading this disclosure are deemed within the spirit and scope of this invention.

C L A I M S

What is claimed is:

1. A non-volatile semiconductor mass storage device comprising:

a. a plurality of non-volatile storage blocks, wherein each block is selectively programmable and erasable wherein only blocks containing no data are programmed;

b. means for determining whether any unprogrammed blocks remain;

c. means for replacing superseded data with updated data including nonvolatile flag means which are set for temporarily ignoring blocks having superseded data and programming means for storing updated data into a block containing no data; and

d. means for periodically erasing all blocks having flag means which are set,

whereby an erase cycle is not needed each time updated data replaces superseded data.

2. The device according to claim 1 further comprising means for correlating a logical address assigned the superseded data to a physical address of updated data in the device.

3. The device according to claim 2 wherein the flag and the logical address are formed of nonvolatile content addressable memory.

4. The device according to claim 1 wherein the means for periodically erasing simultaneously erases all blocks having flag means which are set.

5. The device according to claim 4 wherein the means for periodically erasing comprises a plurality of nonvolatile single bit storage cells, one cell for and coupled to each block, each cell for storing an appropriate second flag and a plurality of volatile latches, one for each cell, coupled to receive a logic state of the cell during an erase cycle.

6. The device according to claim 5 wherein all latches simultaneously receive the logic state.

7. The device according to claim 3 further comprising means for ensuring no block is subjected to more than a predetermined number of erase cycles than any other block.

8. The device according to claim 7 wherein the means for ensuring comprises a counter for each block for counting each erase cycle to which that block has been subjected.

9. The device according to claim 8 further comprising means for setting a maximum count value coupled to the block, and an erase inhibit flag coupled to the counter having a set condition and an unset condition for each block for preventing further erases to a block having its erase inhibit flag in the set condition.

10. The device according to claim 9 wherein the counter includes a plurality of bits programmed by binary counting.

11. The device according to claim 9 wherein the counter includes a plurality of bits programmed by sequentially programming each bit, one at a time, wherein each programmed bit represents a count of one.

12. The device according to claim 9 wherein the counter and

the erase inhibit flag are formed in the content addressable memory.

13. The device according to claim 9 further comprising a reset means for erasing all counters and all flags for every block at the maximum count value once insufficient blocks remain to store a pending data file.

14. The device according to claim 13 further comprising means for reading the mass storage comprising:

a. means for receiving the logical address of a data file to be read;

b. means for selecting an appropriate physical address which correlates to the logical address of the data file to be read; and

c. means for accessing the data file to be read from the appropriate physical address.

15. The device according to claim 14 wherein the means for selecting an appropriate physical address comprises coupling the logical address to the content addressable memory.

16. A non-volatile semiconductor mass storage device comprising:

a. a plurality of non-volatile storage blocks, wherein each block is selectively programmable and erasable;

b. a first indicating element to provide a first indicia whether each block has been programmed with a data file;

c. a second indicating element to provide a second indicia whether the data file is superseded; and

d. a selecting element to program a new data file into an empty block.

17. The device according to claim 16 further comprising a device to correlate a logical address assigned the superseded data to a physical address of updated data in the device.

18. The device according to claim 17 wherein the first indicating element, the second indicating element and the logical address are formed of nonvolatile content addressable memory.

19. The device according to claim 16 further comprising an erase element to periodically simultaneously erase all blocks having a superseded data file.

20. The device according to claim 19 wherein the erase element include a plurality of nonvolatile single bit storage cells, one cell for and coupled to each block, each cell for storing an appropriate second flag and a plurality of volatile latches, one for each cell, coupled to receive a logic state of the cell during an erase cycle.

21. The device according to claim 20 wherein all latches simultaneously receive the logic state.

22. The device according to claim 18 further comprising a first controller to ensure that no block is subjected to more than a predetermined number of erase cycles than any other block.

23. The device according to claim 22 wherein the first controller includes a counter for each block for counting each erase cycle to which that block has been subjected.

24. The device according to claim 23 wherein the counter

includes a plurality of bits programmed by binary counting.

25. The device according to claim 23 wherein the counter includes a plurality of bits programmed by sequentially programming each bit, one at a time, wherein each programmed bit represents a count of one.

26. The device according to claim 23 further comprising a second controller to set a maximum count value coupled to the block, and an erase inhibit flag coupled to the counter having a set condition and an unset condition for each block for preventing further erases to a block having its erase inhibit flag in the set condition.

27. The device according to claim 26 wherein the counter and the erase inhibit flag are formed in the content addressable memory.

28. The device according to claim 26 further comprising a reset element to erase all counters, first indicating element, second element and all flags for every block at the maximum count value once insufficient blocks remain to store a pending data file.

29. The device according to claim 28 further comprising a reading element for the mass storage comprising:

a. a first circuit to receive the logical address of a data file to be read;

b. a second circuit to select an appropriate physical address which correlates to the logical address of the data file to be read; and

c. a third circuit to access the data file to be read from the appropriate physical address.

30. The device according to claim 29 wherein the second circuit couples the logical address to the content addressable memory.

31. A non-volatile semiconductor mass storage device comprising:

a. a plurality of non-volatile storage blocks, wherein each block is selectively programmable to store data and is selectively erasable;

b. a plurality of first flags, one first flag for each block, each first flag having a first logic state to indicate that a block has not been programmed with data and a second logic state to indicate that the block has been programmed with data;

c. a selecting element to identify an empty block having a first flag in the first logic state to receive new data;

d. a plurality of second flags that can only be changed in a block having its first flag in the second logic state, one second flag for each block, each second flag having a third logic state to indicate that the data in a block is valid and a fourth logic state to indicate that the data in the block has been superseded; and

e. a content addressable memory for correlating a logical address assigned the superseded data to a physical address of updated data in the device.

32. The device according to claim 31 further comprising means for correlating a logical address assigned the superseded data to a physical address of updated data in the device.

33. The device according to claim 31 further comprising means for simultaneously erasing each block having a flag in

the fourth logic state.

34. The device according to claim 33 wherein the means for simultaneously erasing comprises a plurality of nonvolatile single bit storage cells, one cell for and coupled to each block, each cell for storing an appropriate second flag and a plurality of volatile latches, one for each cell, coupled to receive a logic state of the cell during an erase cycle.

35. The device according to claim 34 wherein all latches simultaneously receive the logic state.

36. The device according to claim 35 wherein the storage blocks store data in flash memory cells.

37. The device according to claim 36 wherein the flags are stored in flash memory cells.

38. The device according to claim 37 wherein the means for correlating are stored in flash memory cells.

39. The device according to claim 35 wherein the storage blocks store data in E^2PROM cells.

40. The device according to claim 39 wherein the flags are stored in E^2PROM cells.

41. The device according to claim 40 wherein the means for correlating are stored in E^2PROM cells.

42. A non-volatile semiconductor mass storage device comprising:

a. a plurality of non-volatile storage blocks, wherein each block is selectively programmable to store data

and is selectively erasable;

b. a plurality of first flags, one flag for each block;

c. means for setting a corresponding first flag to indicate that a corresponding block has been programmed with data;

d. means for selecting a block having an unset first flag for storing a new data;

e. means for programming the new data into the block having an unset first flag;

f. a plurality of second flags, one flag for each block;

g. means for setting a corresponding second flag for a programmed block to indicate that data in the programmed block has been superseded;

h. a content addressable memory having an entry for each block for correlating a logical address assigned the superseded data to a physical address of updated data in the device; and

i. means for erasing each block having a set second flag once too few blocks remain for storing a next new data file, including each appropriate content addressable memory entry, first flag and second flag corresponding to each such block.

43. A method of storing data into a non-volatile semiconductor mass storage device having a plurality of non-volatile storage blocks, wherein each block is selectively programmable and erasable wherein only blocks containing no data are programmed, the method comprising the steps of:

a. determining whether any unprogrammed blocks remain;

b. replacing superseded data with updated data including flag means which are set for temporarily ignoring blocks having superseded data and programming means for storing updated data into a block containing no data without erasing the superseded data; and

c. periodically erasing all blocks having flag means which are set.

44. The method according to claim 43 further comprising correlating a logical address assigned the superseded data to a physical address of updated data in the device.

45. The method according to claim 44 wherein the step of periodically erasing simultaneously erases all blocks having flag means which are set.

46. The method according to claim 45 wherein the step of periodically erasing includes the step of simultaneously loading volatile latches from the flag means.

47. The method according to claim 46 further comprising the step of ensuring no block is subjected to more than a predetermined number of erase cycles than any other block.

48. The method according to claim 47 further comprising reading the mass storage comprising the steps of:

a. receiving the logical address of a data file to be read;

b. selecting an appropriate physical address which correlates to the logical address of the data file to be read; and

c. accessing the data file to be read from the appropriate physical address.

49. A method of storing data into a non-volatile semiconductor mass storage device having a plurality of non-volatile storage blocks, wherein each block is selectively programmable to store data and is selectively erasable, the method comprising:

a. determining whether a block has been programmed with data;

b. providing a first indicia for each block which has been programmed with data;

c. programming new data into a block which has no first indicia; and

d. if the new data supersedes old data stored in a block, providing a second indicia to the block storing the old data.

50. A method of storing data into a non-volatile semiconductor mass storage device having a plurality of non-volatile storage blocks, wherein each block is selectively programmable to store data and is selectively erasable, the method comprising:

a. setting a corresponding one of a plurality of first flags, one flag for each block, to indicate that a corresponding block has been programmed with data;

b. selecting a block having an unset first flag for storing a new data;

c. programming the new data into the block having an unset first flag;

d. setting a corresponding one of a plurality of second flags, one flag for each block, for a programmed block to indicate that data in the programmed block has been superseded; and

e. erasing each block having a set second flag once too few blocks remain for storing a next new data, including the first flag and the second flag corresponding to each such block.

51. A nonvolatile memory integrated circuit device comprising:

a. a plurality of storage cells which are segmented into a plurality of memory blocks of a predetermined size;

b. a plurality of latches, one latch coupled to each block for coupling an erase voltage to the block; and

c. a control element to simultaneously set all latches for erasing a predetermined set of the blocks.

52. The memory according to claim 51 further comprising a plurality of nonvolatile memory bits, one memory bit for each block for storing a data bit representative of the

predetermined set, each memory bit coupled to one of the latches wherein during an erase cycle a control signal couples each data bit into each appropriate latch.

53. The memory according to claim 52 wherein the memory bit for each latch in a set condition is erased simultaneously with its associated block during the erase cycle.

54. The memory according to claim 53 wherein each memory bit is a flash memory bit.

55. The memory according to claim 53 wherein each memory bit is an E^2PROM memory bit.

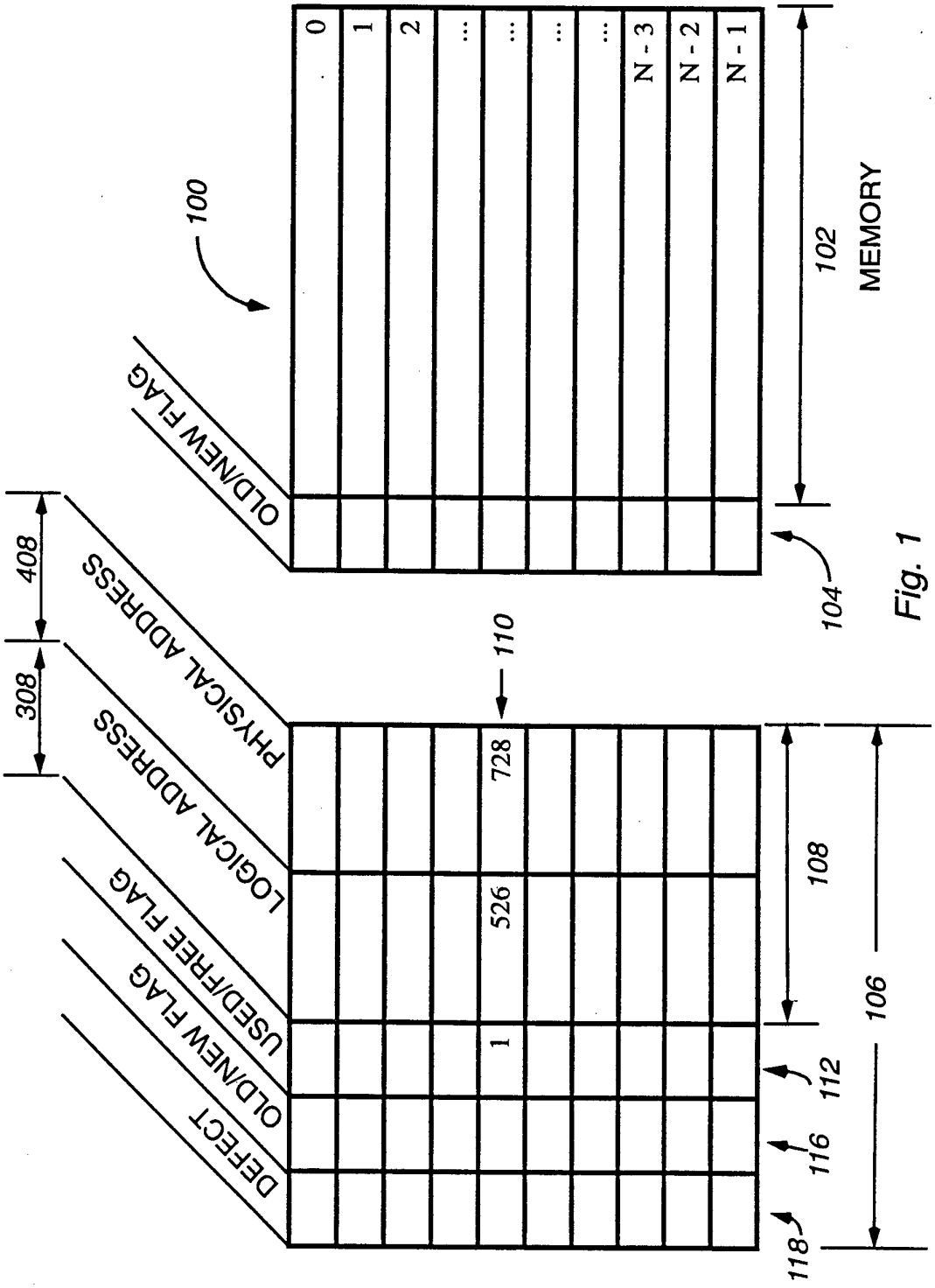

100
0
1
2
...
...
...
...
N - 3
N - 2
N - 1
102
MEMORY
OLD/NEW FLAG
104
110
408
308
PHYSICAL ADDRESS
LOGICAL ADDRESS
USED/FREE FLAG
OLD/NEW FLAG
DEFECT
728
526
1
108
106
112
116
118

Fig. 1

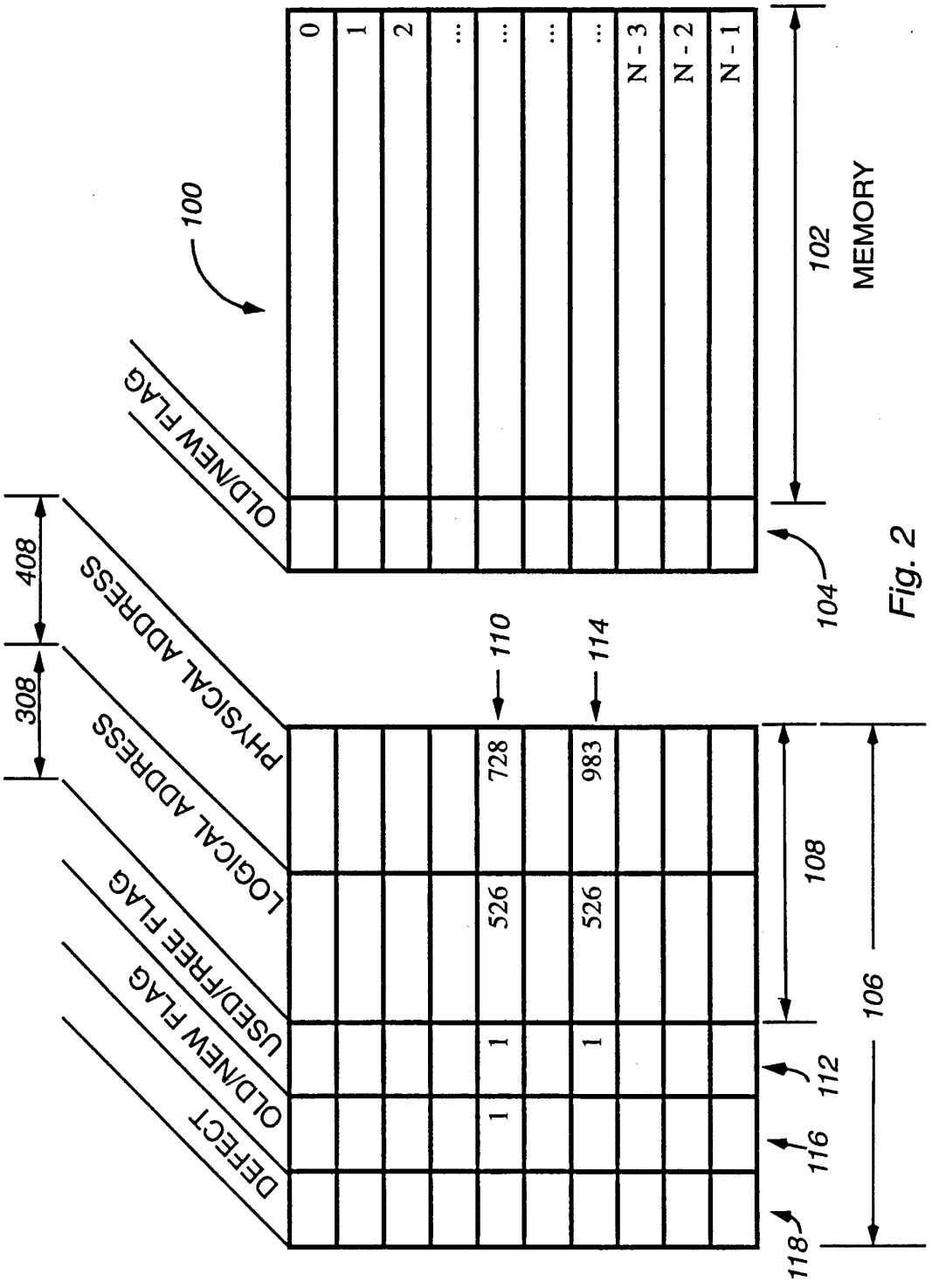

100
0
1
2
...
...
...
...
N - 3
N - 2
N - 1
102
MEMORY
OLD/NEW FLAG
104
408
308
PHYSICAL ADDRESS
LOGICAL ADDRESS
USED/FREE FLAG
OLD/NEW FLAG
DEFECT
110
114
728
983
526
526
1
1
1
108
106
112
116
118

Fig. 2

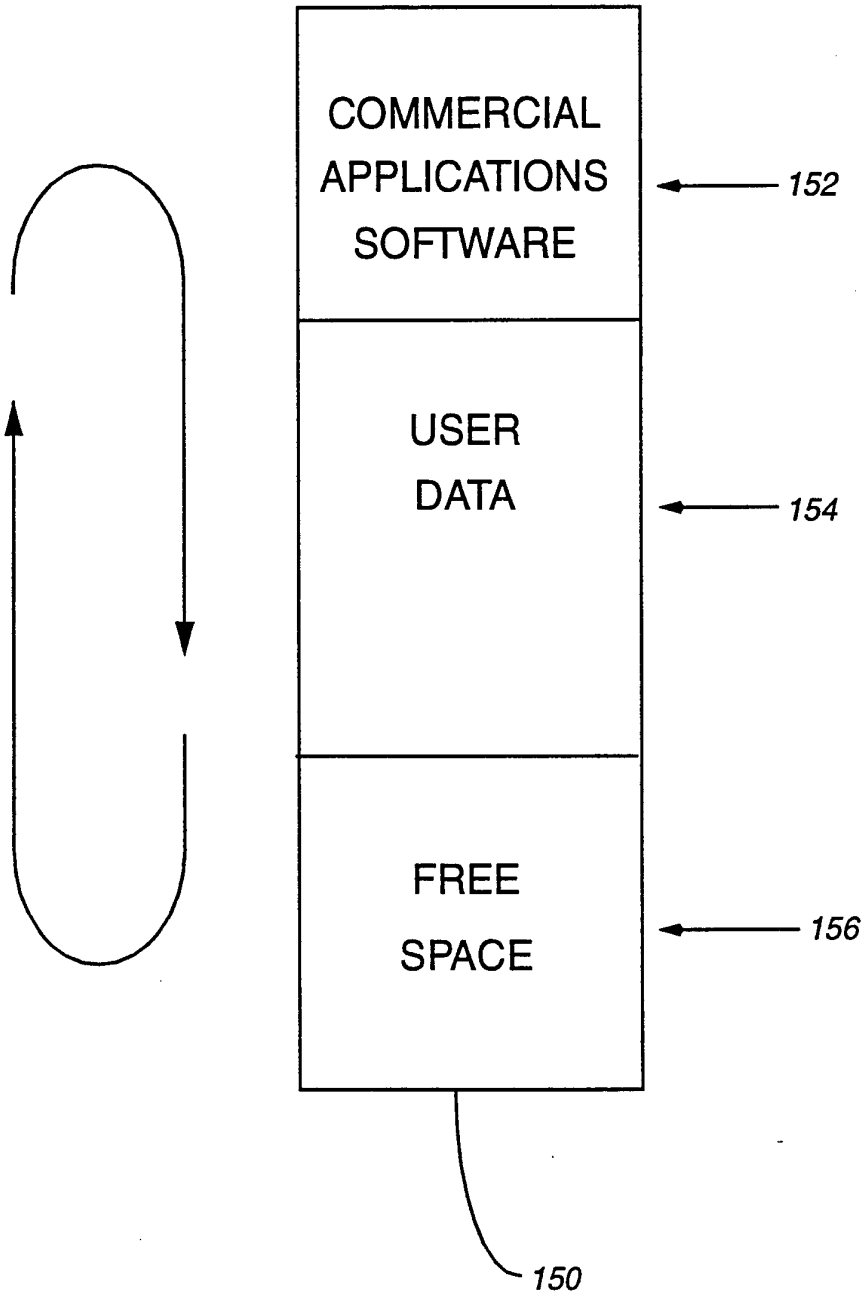

COMMERCIAL APPLICATIONS SOFTWARE
152
USER DATA
154
FREE SPACE
156
150

Fig. 3

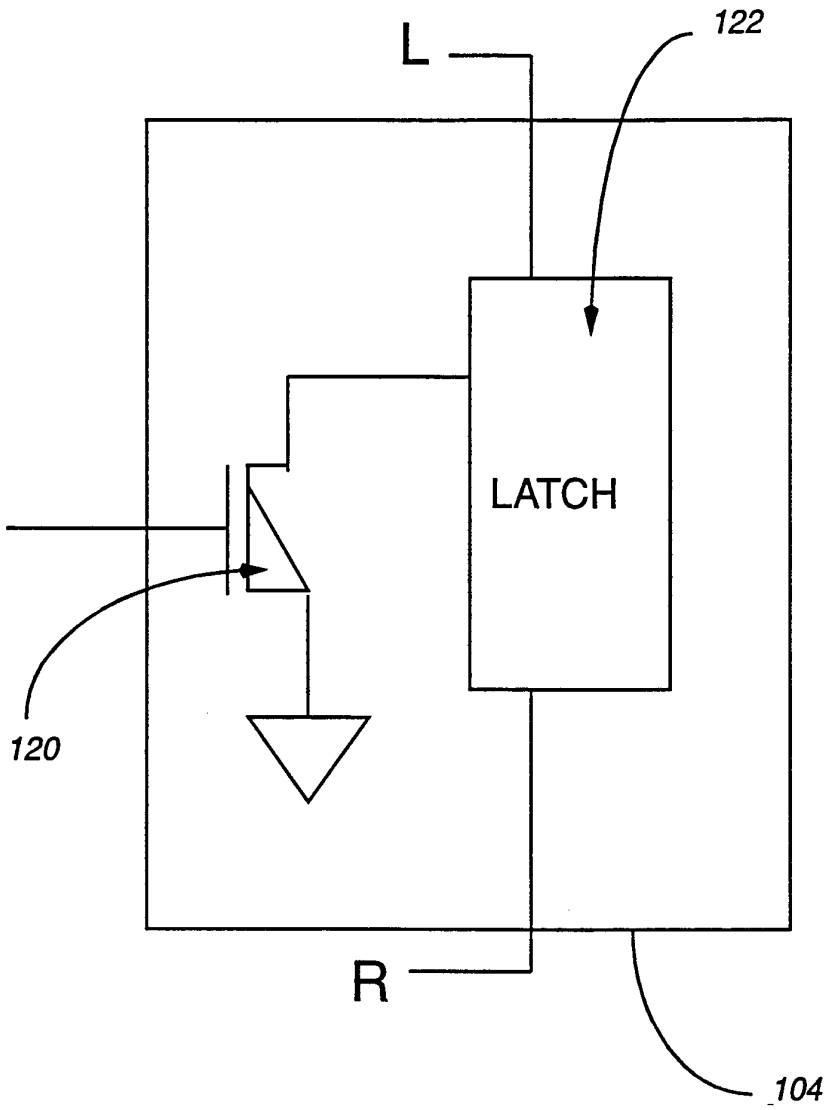


Fig. 4

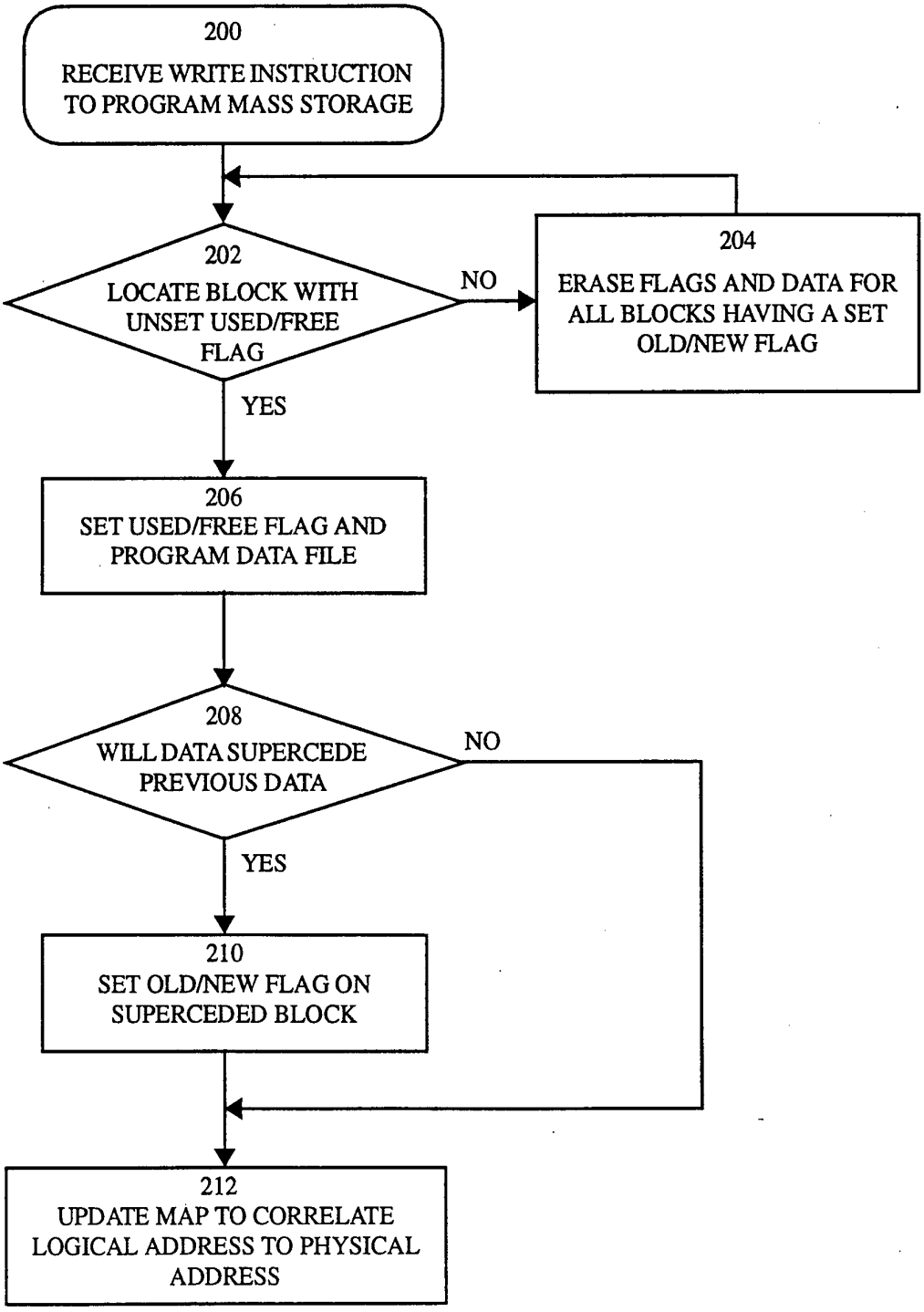

200
RECEIVE WRITE INSTRUCTION TO PROGRAM MASS STORAGE
202
LOCATE BLOCK WITH UNSET USED/FREE FLAG
NO
204
ERASE FLAGS AND DATA FOR ALL BLOCKS HAVING A SET OLD/NEW FLAG
YES
206
SET USED/FREE FLAG AND PROGRAM DATA FILE
208
WILL DATA SUPERCEDE PREVIOUS DATA
NO
YES
210
SET OLD/NEW FLAG ON SUPERCEDED BLOCK
212
UPDATE MAP TO CORRELATE LOGICAL ADDRESS TO PHYSICAL ADDRESS

Fig. 5

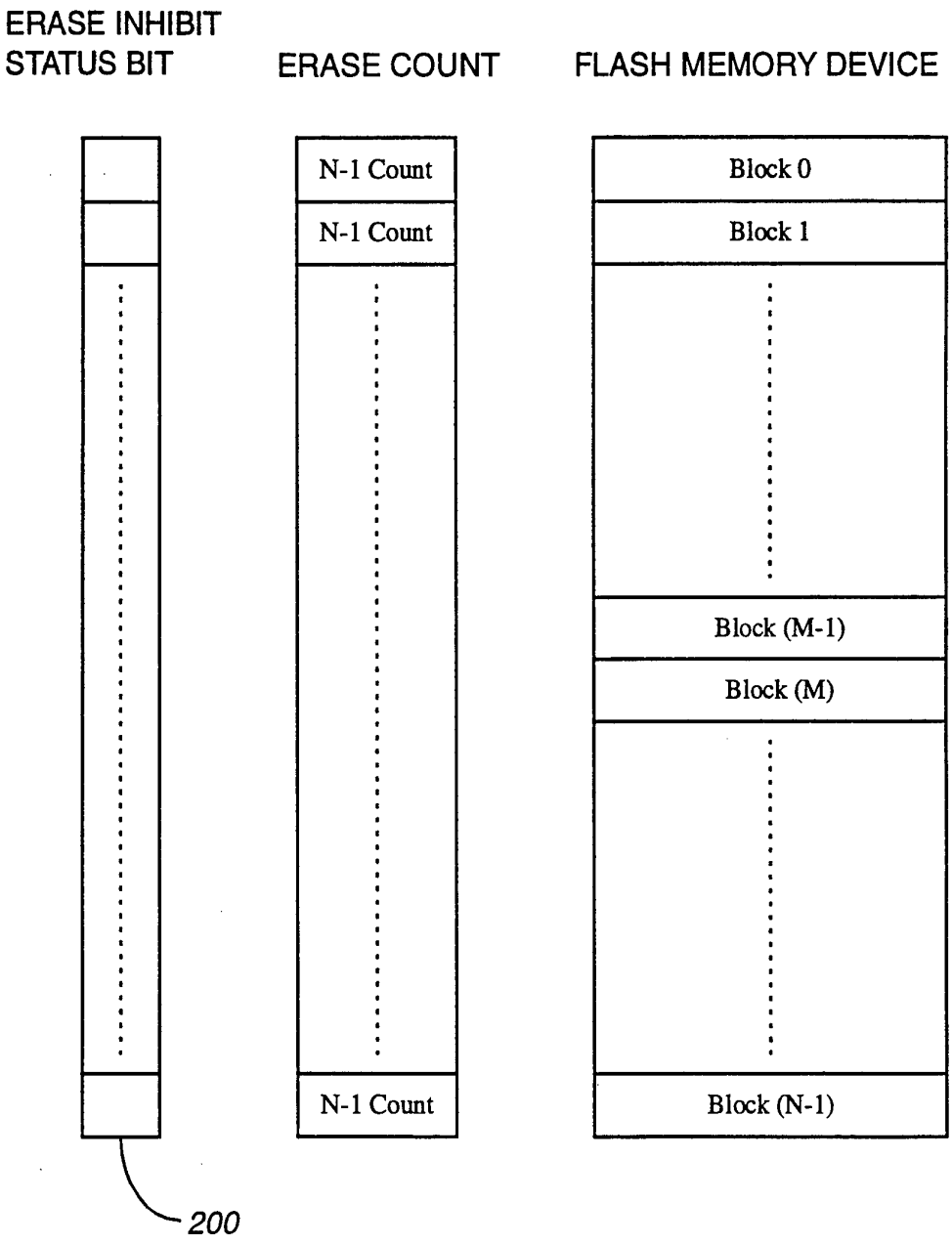

ERASE INHIBIT STATUS BIT
ERASE COUNT
FLASH MEMORY DEVICE
N-1 Count
N-1 Count
N-1 Count
Block 0
Block 1
Block (M-1)
Block (M)
Block (N-1)
200

Fig. 6

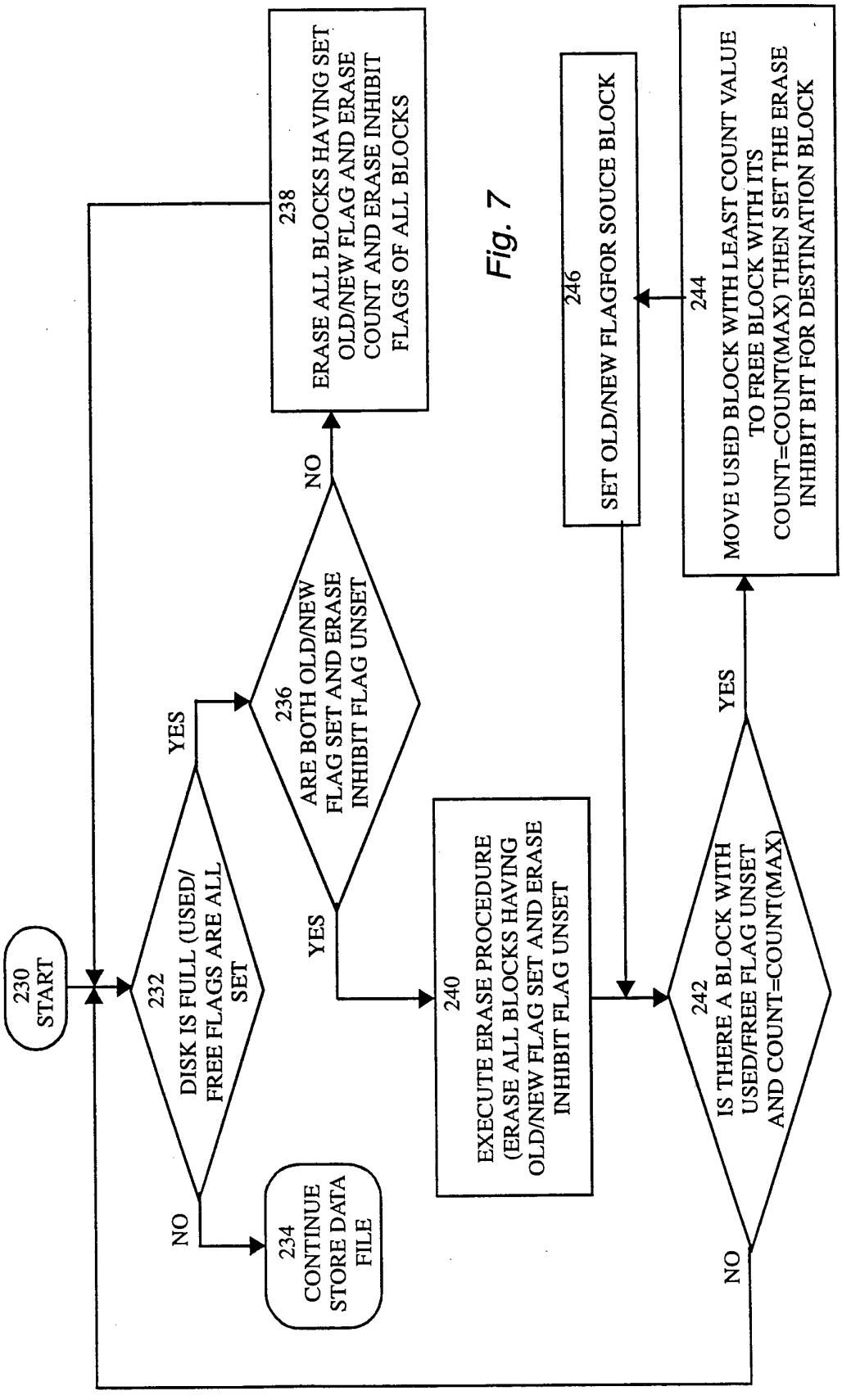

230
START
232
DISK IS FULL (USED/
FREE FLAGS ARE ALL
SET
NO
234
CONTINUE
STORE DATA
FILE
YES
236
ARE BOTH OLD/NEW
FLAG SET AND ERASE
INHIBIT FLAG UNSET
NO
238
ERASE ALL BLOCKS HAVING SET
OLD/NEW FLAG AND ERASE
COUNT AND ERASE INHIBIT
FLAGS OF ALL BLOCKS
YES
240
EXECUTE ERASE PROCEDURE
(ERASE ALL BLOCKS HAVING
OLD/NEW FLAG SET AND ERASE
INHIBIT FLAG UNSET
242
IS THERE A BLOCK WITH
USED/FREE FLAG UNSET
AND COUNT=COUNT(MAX)
NO
YES
244
MOVE USED BLOCK WITH LEAST COUNT VALUE
TO FREE BLOCK WITH ITS
COUNT=COUNT(MAX) THEN SET THE ERASE
INHIBIT BIT FOR DESTINATION BLOCK
246
SET OLD/NEW FLAGFOR SOUCE BLOCK

Fig. 7

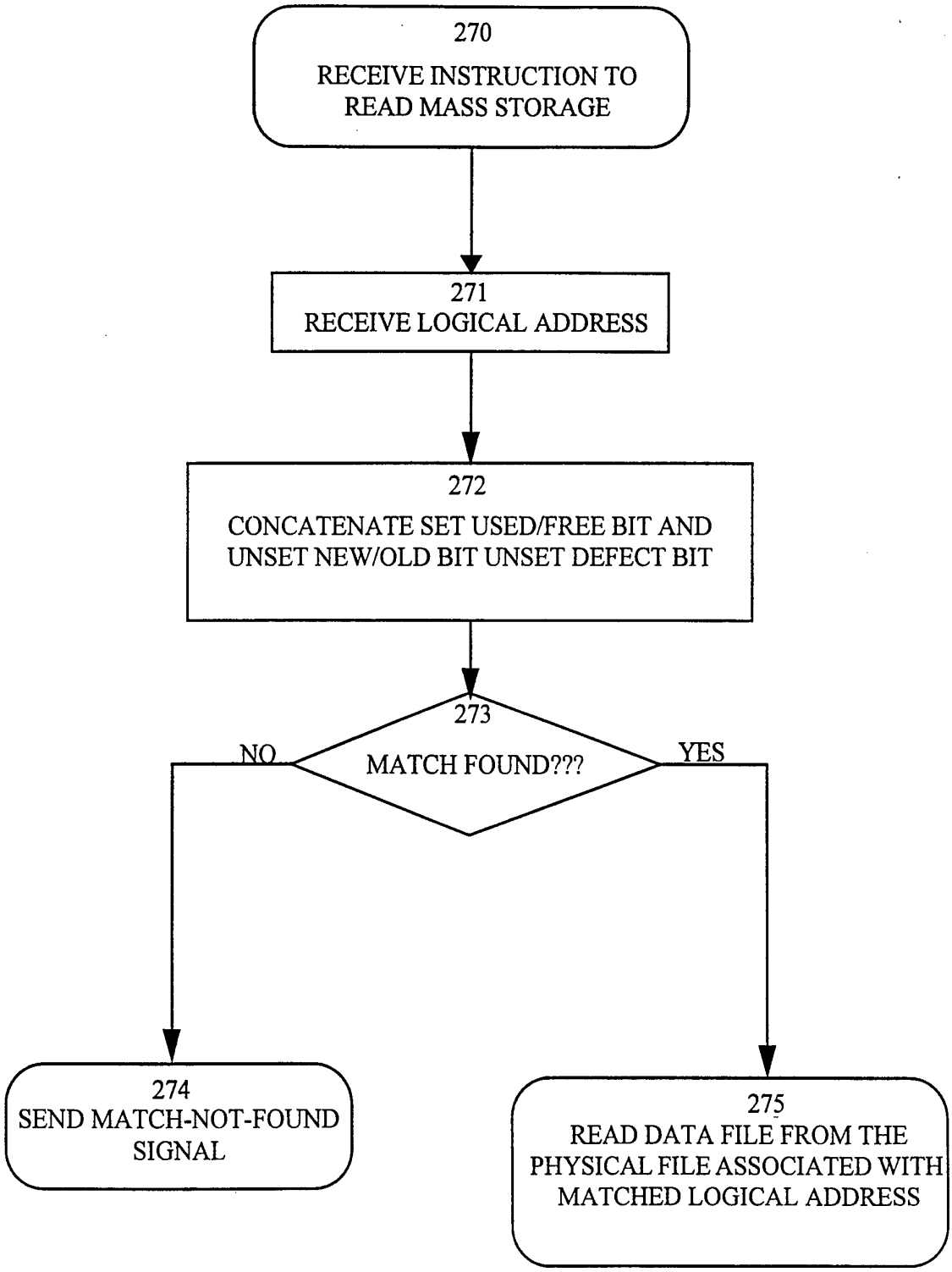

270
RECEIVE INSTRUCTION TO READ MASS STORAGE
271
RECEIVE LOGICAL ADDRESS
272
CONCATENATE SET USED/FREE BIT AND UNSET NEW/OLD BIT UNSET DEFECT BIT
273
MATCH FOUND???
NO
YES
274
SEND MATCH-NOT-FOUND SIGNAL
275
READ DATA FILE FROM THE PHYSICAL FILE ASSOCIATED WITH MATCHED LOGICAL ADDRESS

Fig. 8

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/03168

A. CLASSIFICATION OF SUBJECT MATTER

IPC(5) :G06F 12/02
US CL :395/425;365/189.05
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/425; 365/189.05

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS, Dialong

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	JP, A, 62-283496 (Canon Inc.) 09 December 1987 See Purpose and Constitution.	1-4, 7-12, 16-19. 22-27, 31-33
Y	JP, A, 62-683497 (Canon Inc.) 09 December 1987 See Purpose and Constitution	1-4, 7-12, 16-19, 22-27, 31-33, 42-45 & 49-50
Y	JP, A, 59-45695 (FUJITSU k.k) 14 March 1984, See Purpose and Constitution	1-4, 7-12, 16-18, 22-27, 31-33, 42-45 & 49-50

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be part of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
13 JULY 1994	18 AUG 1994
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. NOT APPLICABLE	Authorized officer DAVID ROBERTSON Telephone No. (703) 305-3825

Form PCT/ISA/210 (second sheet)(July 1992)*

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/03168

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
P	US, A, 5,270,979 (Harari et al) 14 December 1993 See Abstract	1-55
E	US. A, 5,303,198 (Adadii et al) 12 April 1994 See Abstract	1-55
Y	Computer Architectue And Parallel Processing, K. Hwang & F.A. Briggs, 1984 McGraw-Hill, Inc. p.64	1-4, 7-12, 16-19, 22-27, 31-33, 42-45 and 49-50

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US94/03168

Box I Observations where certain claims were found unsearchable (Continuation of item 1 of first sheet)

This international report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:
(Form PCT/ISA/206 Previously Mailed.)

Group I, claims 1-50 drawn to tracking the use of memory locations, classified in class 395, subclass 425.

Group II, claims 51-55, drawn to the specifics of a group of latches, classified in class 365, subclass 189.05.

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest.

☐ No protest accompanied the payment of additional search fees.