

PATENT REQUEST: STANDARD PATENT/PATENT OF ADDITION

We, being the person(s) identified below as the Applicant, request the grant of a patent to the person identified below as the Nominated Person, for an invention described in the accompanying standard complete specification.
Full application details follow.

(71) Applicant *ALCATEL N.V.*

Address *Strawinskylaan 341, 1077 XX, Amsterdam, The Netherlands*

(70) Nominated Person *ALCATEL N.V.*

Address *Strawinskylaan 341, 1077 XX, Amsterdam, The Netherlands*

(54) Invention Title - **"METHOD OF UPDATING DATA STORED IN A STORAGE UNIT"**

(72) Name(s) of actual inventor(s) **Lorenzo LISCA; Murizio BARILLARI;**

(74) Address for service in Australia - **Patent Department, Alcatel Australia Limited,
280 Botany Road, Alexandria, 2015, N.S.W. Australia**

ASSOCIATED PROVISIONAL APPLICATION(S) DETAILS

(60) Application Number(s) and Date(s)

BASIC CONVENTION APPLICATION(S) DETAILS

(31) Application Number	(33) Country	Country Code	(32) Date of Application
M191A003066	ITALY	IT	15 November 1991

DIVISIONAL APPLICATION DETAILS

(62) Original application number:


PARENT INVENTION DETAILS (Patent of Addition requests only)

(61) Application Number Patent Number

Drawing number recommended to accompany the abstract..

ALCATEL N.V.

F038742 05/11/92


(Signature)

November 2nd, 1992

P/00/008
Section 29(1)
Regulation 3.1(2)

AUSTRALIA


Patents Act 1990

NOTICE OF ENTITLEMENT

We, **ALCATEL N.V.**
of Strawinskylaan 341, 1077 xx Amsterdam, The Netherlands,
being the applicant in respect of Application No.
for an invention entitled "METHOD OF UPDATING DATA STORED IN A STORAGE
UNIT" described in the accompanying specification, state the following:

1. The company nominated for the grant of the patent has entitlement from the actual inventor by mesne assignment.
2. The company nominated for the grant of the patent has entitlement from the applicant of the basic application listed on the patent request form by assignment.
3. The basic application listed on the request form is the first application made in a Convention country in respect of the invention.

ALCATEL N.V.



B. O'CONNOR

November 2nd, 1992



AU9228150

(12) PATENT ABRIDGMENT (11) Document No. AU-B-28150/92
(19) AUSTRALIAN PATENT OFFICE (10) Acceptance No. 664718

(54) Title
METHOD OF UPDATING DATA STORED IN A STORAGE UNIT

International Patent Classification(s)
(51)⁵ **G11C 007/00 G11C 016/02**

(21) Application No. : **28150/92** (22) Application Date : **05.11.92**

(30) Priority Data

(31) Number (32) Date (33) Country
M191A3066 15.11.91 IT ITALY

(43) Publication Date : **20.05.93**

(44) Publication Date of Accepted Application : **30.11.95**

(71) Applicant(s)
ALCATEL N.V.

(72) Inventor(s)
LORENZO LISCA; MURIZIO BARILLARI

(74) Attorney or Agent
ALCATEL AUSTRALIA LIMITED , BOX 525 G.P.O., SYDNEY N.S.W. 2001

(56) Prior Art Documents
JP 59-71180
JP 58-9291

(57) Claim

1. A method of updating data stored in storage locations of a flash eprom storage unit, wherein updating information is loaded into unoccupied storage locations, said storage locations containing said stored data being updated at periodic or occasional intervals by using said loaded updating information.

P/00/011 28/5/91

Regulation 3.2

664718

AUSTRALIA
Patents Act 1990

**ORIGINAL
COMPLETE SPECIFICATION
STANDARD PATENT**

Invention Title:

"METHOD OF UPDATING DATA STORED IN A STORAGE UNIT"

The following statement is a full description of
this invention, including the best method of
performing it known to us:-

This invention relates to a method of updating data stored in storage locations of a storage unit, in particular of a Flash EPROM.

Updating of data stored in some types of storage units, particularly non-volatile
5 memories, is a problem because it is a time consuming activity and often quasi-independent from the amount of updating to be carried out.

This is the case, for example, of disks for which access time is mainly due to mechanical movements of heads. Each disk reader includes a logic circuit for controlling the disk write physical operations. Normally as soon as a writing or updating
10 command is received, a write physical operation occurs. This method normally involves an enormous waste of time due to the fact that consecutive write logic operations correspond to consecutive write physical operations on different and often distant physical areas of the disk, and therefore a long time is consumed for moving heads from one disk area to another.

The same problem arises with Flash EPROMs. Flash EPROMs are a special type
15 of EEPROMs whose cost is remarkably lower than conventional EEPROMs and moreover the attainable integration degree on a chip is higher (256 kbytes per chip against 32 kbytes per chip).

The main difference between the two types of device lies in that Flash EPROMs
20 must be erased totally (like UV EPROMs), ie. erasure concerns the entire chip, by a simple electrical command, whilst EPROMs are erased on a byte by byte basis; both types of memories are written on a byte by byte basis.

An important feature of Flash EPROMs lies in that it is possible to change the value of any bit from 1 to 0 (programming operating) but it is necessary to erase the entire memory device (erase operation) in order to change the value of a bit from 0 to 1.
25 The erase operation goes on for some seconds and consists in turning to 1 all bits of the memory device; Flash EPROM manufacturers, eg. Advanced Micro Devices, recommend to turn, via software or firmware, all the bits of the memory device to 0 before executing such erasing operation in such a way that the locations of the memory
30 device are "consumed" uniformly.

Therefore, when it is necessary to change even a single bit from 0 to 1, first the content of the entire Flash EPROM has to be transferred to a RAM having at least the same capacity, then the value of the bit is to be changed, after which the erasing action is carried out and finally the content of the RAM is to be copied again in the Flash
35 EPROM. It is clear that such a method is time consuming and this is particularly

disadvantageous especially when the content of the memory device has to be changed frequently.

Should one often change the content of single or few bytes, the use of such method requires not to exceed the maximum limit of erasure/write cycles (1000 to
5 10000) warranted as to be error-free for such type of device, which is a further problem.

A known arrangement assures an error-free operation for an EEPROM subjected to a number of erasure/write cycles greater than those allowed for individual locations of EEPROM itself.

The known arrangement is based upon a store having a capacity greater than the
10 number of memory zones (comprising several bytes) each time necessary for data to be stored at the same time, and consists first of all in storing data in a single memory zone and then in switching, according to a suitable criterion, on a successive memory zone under the control of a memory controller, and use this successive memory zone until such suitable criterion suggests otherwise; hence it is substantially a question of
15 operative environment change.

The method of the known arrangement requires memories of much higher capacity than really needed and moreover it does not solve the problem of quick updating the stored data.

It is an object of the present invention to overcome the drawbacks of prior art
20 methods.

According to the invention there is provided a method of updating data stored in storage locations of a flash eprom storage unit, wherein updating information is loaded into unoccupied storage locations, said storage locations containing said stored data being updated at periodic or occasional intervals by using said loaded updating
25 information.

A remarkable reduction in updating time can be attained by writing information relative to updating to be carried out into unoccupied storage locations and executing the updating all at once on the storage locations containing data.

The invention will now be described with reference to the accompanying
30 drawings, in which:

Figure 1 shows the structure of a storage unit and a possible storage allocation for updating information according to the present invention;

Figure 2 shows the structure of a storage unit and another possible storage allocation for updating information according to the present invention;

35 Figures 3 to 7 show a possible evolution in time of the content of storage locations containing data and updating information, according to the present invention;



and

Figures 8 to 11 show another possible evolution in time of the content of storage locations containing data and updating information, according to the present invention.

In the following, reference will be always made to the case in which the method according to this invention is applied to Flash EPROMs, but the same applies to, eg. disks.

With reference to Figures 1 and 2, UM stands for a generic storage unit. Such storage unit UM is organised in storage locations LM. Data are normally written into the memory starting either from the smallest address, as illustrated in Figures 1 and 2, or from the greatest one, and constitute the stored data DM. Consecutive storage locations LM are often used, but in general the control of memory allocation is entrusted to software or firmware.

Stored data DM are updated, either by software or firmware, by sending write commands, more properly "overwrite" ones, consisting of the data address to be updated and the new binary value of the data itself, to the storage unit UM.

As mentioned, to make storage unit UM execute write commands as they are received can be very inefficient, as in case of disks, or it may not be possible, as in the case of Flash EPROMs where an "overwrite" generally involves a total erasure of the device.

A high gain in efficiency is obtained by loading updating information IA into unoccupied, ie. virgin, storage locations LM. Such updating information IA generally consists of a number of updatings, stored data DM, accumulated in course of time.

The loaded updating information IA can be read out, either occasionally or periodically, and used for updating effectively the stored data DM which such updating information IA is referred to.

Naturally, once updating information IA has been applied to stored data DM, the storage locations LM used to contain it can be reset for future use.

These three operations i.e. loading, updating and reset can be repeated.

In Figure 1 there is illustrated the case wherein updating information IA is loaded into storage locations LM distinct from storage location LM used for stored data DM.

In Figure 1 there is illustrated the case wherein updating information IA is loaded into storage locations LM distinct from storage location LM used for stored data DM.

In Figure 2, on the contrary, updating information IA and stored data DM alternate in the storage locations LM.

Besides the choice of the physical dislocation of updating information IA it is

necessary to make always a choice of the logical dislocation of such information.

A first possibility, illustrated in Figure 1, is to write consecutively all updating information in a well defined storage area independently from the fact that they refer to well defined storage locations. In this instance the updating information shall contain at least a first sequence of bits containing the address of the storage location to be updated, and a second sequence of bits containing the new binary value of the storage location to be updated.

When it is desired to update several consecutive storage locations, a third sequence of bits, containing the number of storage locations in which the content is to be updated, is also needed; the first sequence of bits correspond now to starting address of consecutive storage locations and the second sequence of bits will be the new global binary value.

Updating information IA generally will include a plurality of such sequences of bits.

A second possibility, illustrated in Figure 2, is to dedicate to a first storage area AM1 a second storage area AM2 capable of containing updating information IA relative to the first storage area AM1. Naturally several pairs of such storage areas may be contemplated. In this case the updating information shall contain only a sequence of bits corresponding to the new binary value of the first storage area AM1.

Whatever the physical and logical arrangement of updating information IA is, the advantage in time lies in that loading of updating information IA into unoccupied storage locations is not a time consuming operation and the total erasure of the device occurs only when a number, pre-established by software or firmware, of updatings have to be executed, or at regular intervals, contrary to what was contemplated in the methods used up to now wherein an erasure for each updating was carried out.

In order to better understand the behaviour of the method according to the present invention two examples of possible evolution in time of the content of storage during an updating phase will be now illustrated. In the following the content of storage locations will be expressed as a decimal number.

The first example refers to Figures 3 to 7. The first storage location is indicated by AM1 and corresponds to the first storage area to be updated. The second storage location is indicated by AM2 and corresponds to the second storage area which has to contain the new binary value. At first both storage locations are virgin and their binary value is FF (see Figure 3), as happens in Flash EPROM memories. As a result of a software evolution the first storage location stores the binary value 1C (see Figure 4).

The evolution of the software involves then the need of updating the binary value of the first storage location to A3. The method according to the present invention teaches that such new binary value shall not overwrite the old one but it must be loaded into the second storage location, see Figure 5. At the discretion of software or firmware,
 5 data 1C stored will be updated with the new one A3 (see Figure 6) and then the second storage location will be reset to its initial binary value FF, see Figure 7.

In the special case of Flash EPROMs, when software or firmware decides at a precise moment to apply updatings, it copies the content of the entire Flash EPROM memory into a RAM memory, usually identical to the first one; therefore Figure 5
 10 represents at the same time the content of two storage locations of Flash EPROM and RAM. The content A3 of the second storage location, of RAM, is copied into the first storage location of RAM, as illustrated in Figure 6. Subsequently, the content of the second storage location of RAM, is reset to FF, as shown in Figure 7.

At this point the Flash EPROM is erased completely and then is programmed by using the content of the RAM memory. After such operation, the contents of Flash EPROM and RAM will be identical; therefore Figure 7 represents at the same time the
 15 content of two storage locations of Flash EPROM and of RAM.

The second example refers to Figures 8 to 11. The storage unit has one hundred storage locations, each of a byte, arranged as a matrix having ten columns numbered 0, 1, 2, ..., 9, and then rows numbered 00, 10, 20, ..., 90. Stored in the first nine rows are data DM and the updating information IA is loaded in the last row. The address LC of a storage location is obtained by summing numbers corresponding to the row and column of the location itself.
 20

At first, as illustrated in Figure 8, the content of storage locations having address from 00 to 06 is respectively : 1A, 2B, 4C, 7D, A7, B5, CC and these correspond to stored data. In the storage locations dedicated to updating information there is still no data. The evolution of a software involves afterwards the need of updating the binary value of the storage location whose address is 02 to the new binary value AA. The method according to the present invention teaches that such new binary value shall not
 30 overwrite the old one. In the storage locations having address 90 and 91 updating information corresponding to address 02 and to the new binary value AA are then loaded respectively, as shown in Figure 9. Afterwards there is a new need of updating the binary value of the storage location whose address is 06 to the new binary value AB. Such updating information is then loaded into storage locations whose addresses
 35 are 92 and 93, as shown in Figure 10. As already seen in the foregoing example, at the

discretion of software or firmware, the updating information is applied to the stored data. Software or firmware scans the list of the loaded updates starting from storage location 90 and consequently updates the storage locations containing stored data; finally, storage locations 90, 91, 92 and 93 are reset to FF, as shown in Figure 11.

5 As in the preceding example, in the special case of Flash EPROMs, the updating physical operation has to occur on the RAM memory and this involves two copy operations (from Flash EPROM to RAM and from RAM to Flash EPROM) and an operation of total erasure of the Flash EPROM.

10 When a software requests the read out of a data stored in a storage unit UM, it sends a read out command basically consisting in the address of data to be read. In case the method according to the present invention is used for stored data updating operations, it may happen that the content of the storage location corresponding to the data to be read is not yet updated. In order to obtain a correct readout it is the responsibility of the software, or rather of the firmware, to check if there is any
15 updating information loaded into memory.

20 Taking up the two foregoing examples again, in the first instance it will be sufficient to check whether the content of the second storage area is FF, whilst in the second instance it will be necessary to scan the storage locations corresponding to the last row and, more precisely, it will be necessary to check if one or more of locations having even address 90, 92, 94, 96, 98 contain a binary value corresponding to the address of the data to be read and, consequently, to provide software with the most recent binary value.

25 The method according to the present invention may be used with the subscriber data memories of exchanges and "intelligent" telephone concentrators, ie. with local switching capacities, and in general with data storages of computers used for controlling industrial processes.

30 In such applications data storages, or at least a part thereof, are of non-volatile type, particularly the Flash EPROM type, because of their size and cost effectiveness. In such data storages, directories are often stored and used as an access index for other data stored in the data storage.

The importance of such directories is of course very high and it is common practice to contemplate a field in every record, capable of containing information that allows verification if a corruption in the memory occurred: such a field is called check-sum. The binary value of the check-sum depends upon the binary value of data
35 contained in the record itself. As new data are added in the record, the check-sum is

updated in such a way that an error is not raised during the corruption verification phase. In order to avoid useless and long erasures and rewrites of the memory devices, it is of advantage to use the method according to the present invention at every adding of data contemplating, in addition to the first check-sum field, a number of further
5 check-sum fields, eg. one or two for each record, so that further updatings are loaded on successive check-sum fields and that such fields are then periodically reset after the physical updating of the first one.

Another application of the method according to the present invention is concerned with the field of disk readers for computers. As already pointed out,
10 updating of data contained in such types of non-volatile memories is, in general, particularly onerous. If every disk reader is equipped with a further semiconductor non-volatile memory (in such a way as to avoid the risk of loss of data in case of sudden lack of supply), eg. of Flash EPROM or EEPROM type, in which updating information is loaded, and the updating is carried out all at once on the disk itself, a significant time
15 advantage is obtained; during the read out from the disk it will be necessary to verify if in the semiconductor non-volatile memory there is any updating information relative to data to be read out, with a procedure conceptually similar to what is already illustrated.



The claims defining the invention are as follows:

1. A method of updating data stored in storage locations of a flash eprom storage unit, wherein updating information is loaded into unoccupied storage locations, said storage locations containing said stored data being updated at periodic or occasional intervals by using said loaded updating information.
2. A method as claimed in claim 1, wherein once storage locations containing said stored data are updated, the storage locations containing said updating information are reset.
3. A method as claimed in claim 2, wherein said load, said updating and said resetting are repeated.
4. A method as claimed in claim 1, wherein the updating information relative to a first specific storage area are loaded into a second specific storage area.
5. A method as claimed in claim 1, wherein said updating information comprises at least a first sequence of bits corresponding to a new binary value of specific stored data.
6. A method as claimed in claim 5, wherein said updating information further comprises at least a second sequence of bits corresponding to the address of said specific stored data.
7. A method as claimed in claim 6, wherein said updating information further comprises at least a third sequence of bits corresponding to the number of storage locations occupied by said specific stored data.
8. A method as claimed in claim 1, wherein said unoccupied storage locations belong to a different storage unit.
9. A method substantially as herein described with reference to Figures 1 - 11 of the accompanying drawings.

DATED THIS SEVENTH DAY OF AUGUST 1995

ALCATEL N.V.



ABSTRACT

A method of updating data stored in storage locations of a storage unit, in particular of a Flash EPROM.

Updating data stored in some types of non-volatile storage units, such as Flash EPROMs, is time consuming.

A significant reduction in updating time is obtained by writing information relative to updating to be carried out in unoccupied storage locations and executing the updating all at once on the storage locations containing the data.

28150/92

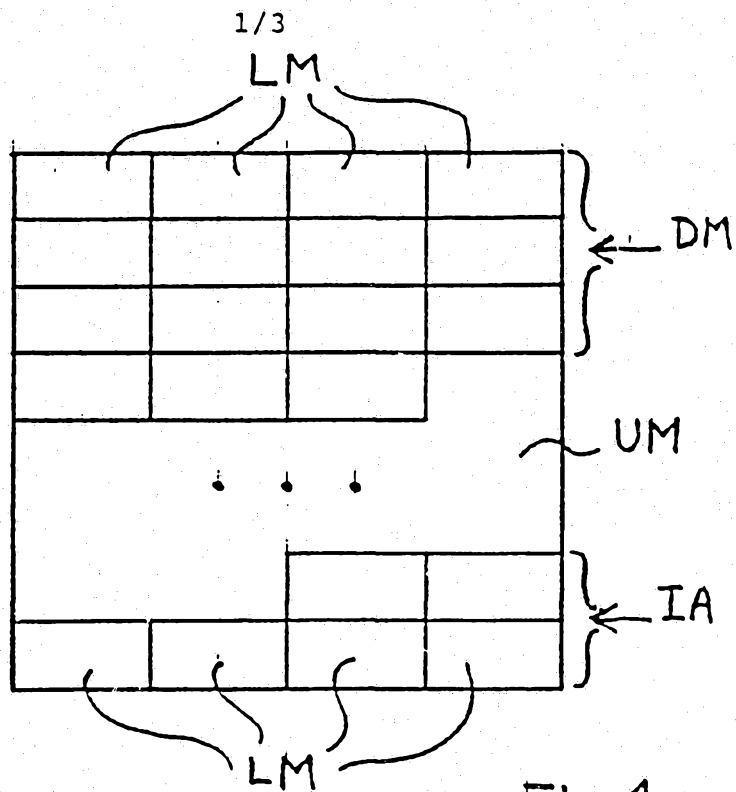


Fig. 1

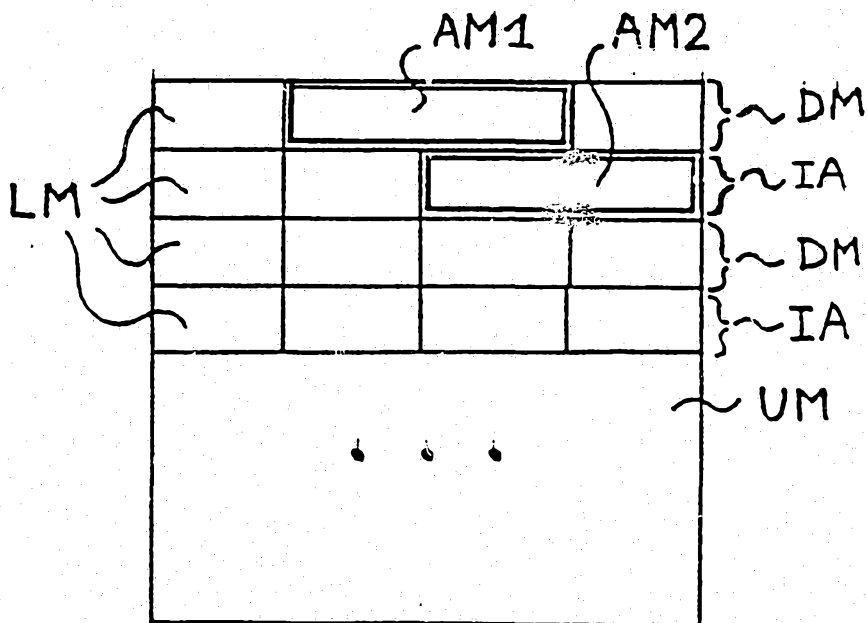
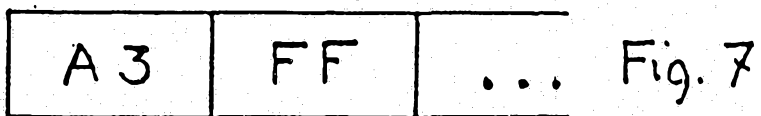
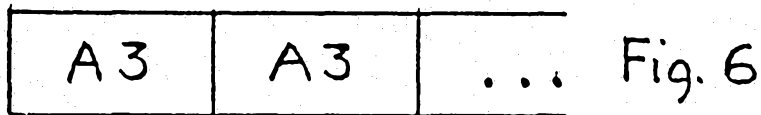
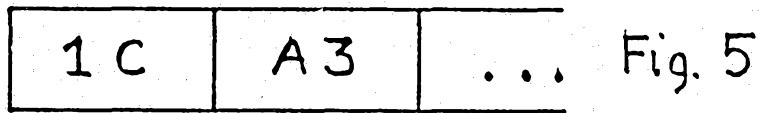
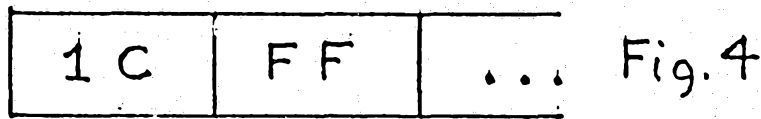
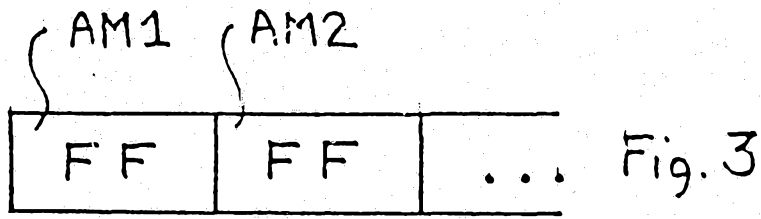


Fig. 2



0	1	2	3	4	5	6	7	8	9	
1A	2B	4C	7D	A7	B5	CC	00
...							10
• • •										
FF	FF	FF	FF			90

Fig. 8

0	1	2	3	4	5	6	7	8	9	
1A	2B	4C	7D	A7	B5	CC	00
...							10
• • •										
02	AA	FF	FF			90

Fig. 9

0	1	2	3	4	5	6	7	8	9	
1A	2B	4C	7D	A7	B5	CC	00
...							10
• • •										
02	AA	06	AB			90

Fig. 10

0	1	2	3	4	5	6	7	8	9	
1A	2B	AA	7D	A7	B5	AB	00
...							10
• • •										
FF	FF	FF	FF			90

Fig. 11