PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G11C 16/06, G06F 11/14

A1

(11) International Publication Number: **WO 94/24673**

(43) International Publication Date: 27 October 1994 (27.10.94)

(21) International Application Number: PCT/GB94/00775

(22) International Filing Date: 13 April 1994 (13.04.94)

(30) Priority Data:
9307623.0 13 April 1993 (13.04.93) GB

(71) Applicant *(for all designated States except US)*: JONHIG LIMITED [GB/GB]; 25 Old broad Street, London EC2 (GB).

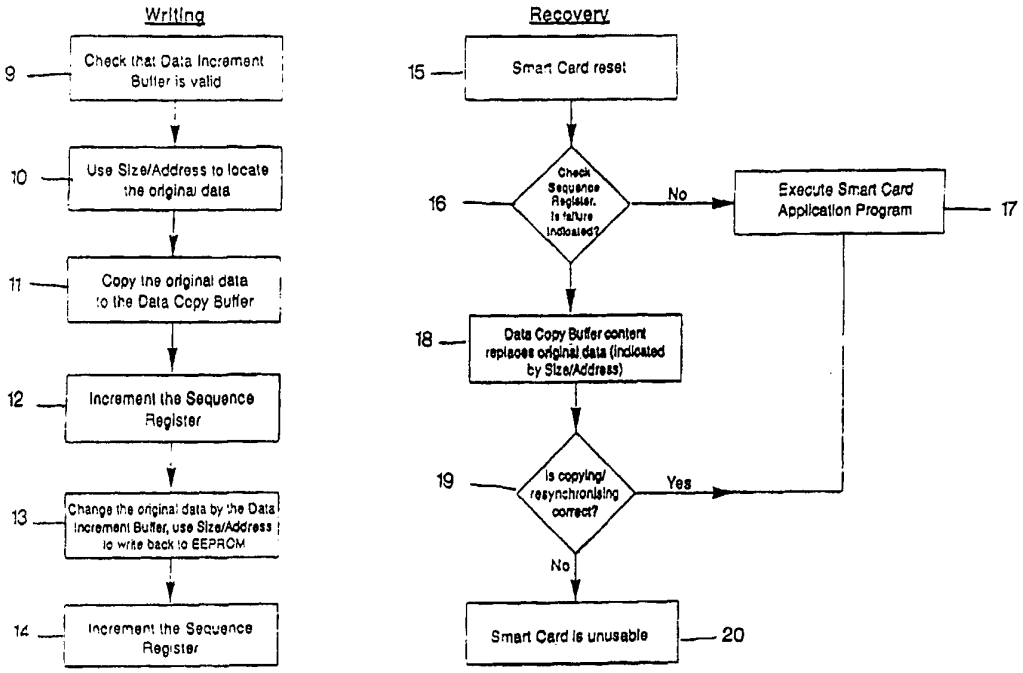
(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: EVERETT, David, B. [GB/GB]; 31 Ashdown Avenue, Brighton, East Sussex BN2 8AH (GB). JACKSON, Keith, M. [GB/GB]; 58 Brighton Road, Shoreham-by-Sea, West Sussex BN43 6RG (GB). MILLER, Ian [GB/GB]; 326 High Street, Dorking, Surrey RH4 1QX (GB).

(74) Agent: SMITH, Martin, Stanley; Stevens, Hewlett & Perkins, 1 St Augustine's Place, Bristol BS1 4UD (GB).

(81) Designated States: AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, ES, FI, GB, HU, JP, KP, KR, KZ, LK, LU, LV, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SK, UA, US, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: DATA WRITING TO NON-VOLATILE MEMORY



(57) Abstract

A method of writing data to non-volatile memory such as electrically erasable programmable read only memory (EEPROM) in a smart card provides a write status region of EEPROM which is examined on each reset of the card. If the preceding write operation was unsuccessful, perhaps because of deliberate manipulation of the card, a recovery procedure is implemented. If recovery is successful the card application can be run. Otherwise the card is unusable.

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				

DATA WRITING TO NON-VOLATILE MEMORY

The invention relates to the writing of data to non-volatile memory. Non-volatile memory is memory which retains data without electrical power being maintained. In particular, the invention relates to the writing of data to memory in transportable integrated circuit devices which are used in conjunction with terminal devices with which they are temporarily coupled for data input and output. An example of such a transportable device is the integrated circuit card (ICC), otherwise known as a "smart card".

Smart cards are coupled by means of an interface to a terminal device whereby power, clock signals, a reset signal and serial data signals may be applied to the card. Generally the interface incorporates a set of electrical contacts for direct temporary electrical connection. However, contactless interfaces employing electromagnetic induction techniques for the application of power have been proposed. In such an arrangement clock, reset and data signals may be coupled electromagnetically or by infra-red or ultra-sonic techniques. Transportable integrated circuit devices may be embodied in tokens of other than card shape. Regardless of shape, such devices will be referred to herein as integrated circuit cards (ICCs). A difficulty with ICCs is that the writing of data to the ICC may be interfered with by disturbing the interface during writing whereby transients or failure in power, reset or clock signals may result in an erroneous write.

A smart card application to which the invention is particularly applicable is in a financial value or "electronic cash" transfer system. Here, data in

smart cards represents value which can be transferred on-line with banks and off-line between cards. Such a system is described in patent applications Nos. WO91/16691 and WO93/08545. It is clearly important in such applications to avoid the effects of erroneous data writing, either accidental or perhaps deliberately instigated by manipulation of power or data lines. The present invention provides a solution.

According to the invention there is provided a method of writing data to non-volatile memory in an integrated circuit device, the device having an interface for temporary connection to a terminal unit; a microprocessor; random access memory and non-volatile memory, the method consisting in allocating a first region of the non-volatile memory for data to be written, allocating a second region of non-volatile memory for write status information to be written, performing a data write operation to write data to said first region, and writing information to said second region signifying a valid data write if, and only if, the data write operation is performed satisfactorily.

In a microprocessor environment there are many copy and write procedures for transferring data and program information between regions of RAM and from RAM to EEPROM, for example, and vice versa. At the operating system level or higher there are usually verification techniques available for verifying the validity of a copy or write operation. This may involve an automatic comparison of the copied or written material with the original or, more usually, the provision of a checksum routine which adds one or more checksum bits to the data which, in accordance with a particular algorithm, provide a link to the data which can be verified to ensure that no write or

copy computation has taken place. If corruption is detected the operation can be repeated until satisfactory. The present invention is not concerned with such techniques and is additional to them, where provided. However, such inbuilt techniques can be used as the basis for determining whether the write operation has been performed satisfactorily in order to write the appropriate information into the said second region of memory. Thus, for example, if data is successfully written to an ICC with inbuilt write verification techniques present then the conclusion of the write process can be taken as indication of a satisfactory write to allow appropriate data to be written to the second region of memory.

The type of non-volatile memory currently used in most smart cards is electrically erasable programmable read-only memory (EEPROM) and the invention is applicable particularly, but not exclusively to this. As far as reading and writing procedures are concerned EEPROM is generally divided into pages and reading or writing is carried out on one page only at a time. It can be expected that a transient writing error may corrupt the contents of one page but not others. Accordingly, it is preferred that the said first and second regions are on different pages.

The invention allows the non-volatile memory to record whether there is an outstanding write error on the device and to take action accordingly when the device is used again, on application of a reset signal. Generally the protocol ISO 7816 is used, which governs the nature of reset, answer-to-reset, power and clock signals etc. If the fault is transient, the reset signal may be applied immediately so that an interrupted transaction may be resumed. If not, the reset signal is applied next time an attempt is made

to use the device. Preferably, in accordance with an aspect of the invention there is provided a method of utilisation of an integrated circuit device to which data has been written as described above, the device including in the non-volatile memory an application program which controls the microprocessor to run a particular application under normal circumstances, the utilisation method including the step of initially reading the said second portion of the non-volatile memory to derive write status information therefrom and, if the write status information indicates an incomplete write operation, by-passing said application program.

Thus, the action effective when an outstanding write error is present on a smart card (for example) may be to render the card useless by continued failure to run the application program. This is software invalidation of the card. Alternatively, a hardware invalidation is possible by providing an overload current to a fuse link in the card, thus blowing the fuse and rendering the card invalid. However, card invalidation is wasteful and preferably the method of utilisation includes, on detection of an incomplete write operation, a procedure of data recovery effective to restore the device to a condition in which the last data write is correct and the status information in the second region of memory reflects this. Should the data recovery procedure fail, then the above-mentioned software or hardware steps of invalidating the card may be taken.

As non-exhaustive examples of the way in which the invention may be used, three specific methods are proposed.

METHOD 1

In accordance with this method it is provided that respective and separate regions of the non-volatile memory are allocated as:-

(a) a sequence register which is said second region of memory;

(b) a data copy buffer;

(c) a size register; and

(d) an address register

and allocating a region of RAM or non-volatile memory as (e) a data incremental buffer, the said first region of non-volatile memory being identified in size and address by data written in memory regions (c) and (d), the said method of writing consisting in:-

1. ensuring that the buffer (e) contains a valid data increment;

2. placing a copy of data to be updated in the buffer (b);

3. incrementing the register (a);

4. incrementing the data at the first region of memory by the amount in buffer (e) and writing the incremental amount to the first region of memory; and

5. incrementing the sequence register (a).

With this method the recovery procedure, when the register (a) indicates recovery is necessary, consists in copying the original (unamended) data from buffer (b) to said first region of memory. This restores the situation to the position before the faulty write operation.

METHOD 2

In this method it is provided that respective and separate regions of the non-volatile memory are

allocated as:-

(f) a write in progress flag register, which is said second region of memory;

(g) a workspace pointer register;

(h) a size register; and

(i) a data pointer register

and allocating a region of RAM or non-volatile memory as (j) a new data pointer register, the said first region of non-volatile memory being identified in size and position by data written in memory regions (g) and (h), the said method of writing consisting in:-

1. setting a workspace pointer in register (g) to the address of non-volatile memory workspace sufficient to hold a contiguous data set corresponding to a size set in register (h);

2. copying to the workspace a copy of new data identified in address by the new data pointer at (j) and in size by the size data at (h);

3. setting the write in progress flag at (f);

4. setting an address in data pointer register (i) to the address of the work-space; and

5. clearing the write in progress flag in register (f).

Here, the recovery procedure comprises repetition of the last two steps (4 and 5), since an error would indicate that the data pointer register had not been properly written.

METHOD 3

In this method it is provided that respective and

separate regions of the non-volatile memory are allocated as:-

(k) a state flag register which is said second region of memory;

(l) a size register;

(m) an address register; and

(n) an update copy buffer

the said first region of non-volatile memory being identified in size and position by data written in registers (l) and (m), the said method of writing consisting in:-

1. copying new data to be written into buffer (n);

2. setting the state flat in register (k);

3. writing said new data to be written to said first region of non-volatile memory; and

4. clearing the state flag in register (k).

Here, new data is typically written directly from RAM and a copy is taken for the update copy buffer (n). If recovery is required, since it is the new data which is held in reserve in (n), the recovery procedure copies this to the required address in EEPROM (for example).

The invention will further be described with reference to the accompanying drawings, of which:-

Figure 1 is a schematic diagram of a smart card having EEPROM organised to effect a first method of data writing and recovery in accordance with the invention;

Figure 2 is a flow diagram in respect of the method used in the card of Figure 1;

Figure 3 is a schematic diagram similar to Figure

1 but in respect of a second method of data writing and recovery in accordance with the invention;

Figure 4 is a flow diagram in respect of the second method;

Figure 5 is a schematic diagram similar to Figures 1 and 3 but in respect of a third method of data writing and recovery in accordance with the invention; and

Figure 6 is a flow diagram in respect of the third method.

Referring to Figure 1 there is shown a smart card 1 which has an interface 2 comprising a set of contacts 3 for making contact with a terminal unit 4. In accordance with the protocol of ISO 7816 the terminal unit provides power, clock signals, a reset signal and serial data signals to the card. The card is an ICC device which includes a microprocessor 5, RAM 6, and EEPROM 7.

The EEPROM 7 is divided into a set of pages 8 and is loaded with an operating system program OS, an application program AP and has a data region DR which holds data which may be read and rewritten.

A first example of the present invention is designated METHOD 1, which is for incremental updating of data in EEPROM. In accordance with this method respective and separate regions of the data region DR of EEPROM are allocated as:-

(a) a sequence register;

(b) a data copy buffer;

(c) a size register; and

(d) an address register.

A region of RAM is allocated as (e) a data incremental buffer, although this could alternatively be in EEPROM also.

Referring now to Figure 2(a) there is shown a flow diagram for the writing of data in accordance with METHOD 1. The steps include:

1. ensuring that the buffer (e) contains a valid data increment (at 9);
2. identifying the EEPROM data to be updated (original data) by reference to the size and address registers (c), (d), giving the original location (at 10);
3. copying the original data to buffer (b) (at 11);
4. incrementing the sequence register (a) (at 12);
5. calculating the new data in RAM by reference to the original data and the data in the data increment buffer (e) and write the new data back to the original location in EEPROM (at 13); and
6. incrementing the register (a) (at 14).

EEPROM is such that its stored data can be corrupted if, whilst the content of the EEPROM is being changed, the power line, or the clock signal are interrupted. With the arrangement described above, data security is provided by the use of the data copy buffer in conjunction with the sequence register. By virtue of internal write verification procedures it can be assumed that if the operating system indicates completion of the write procedure 13 then the written information is in order and the sequence register (a) can be updated appropriately. If the write operation is interrupted by power line or clock signal disruption, for example, then the sequence register remains in its former state which is not appropriate to the attempted write.

In accordance with an aspect of the invention there is a check and recovery procedure available when the card receives the reset signal at any time. Figure 2(b) illustrates this. On reset at 15 the sequence register is checked at 16 to determine whether a write failure is indicated. If not then the application program AP (Figure 1) is executed at 17. If failure is indicated then the original data before the last attempted write operation, which is held in data copy buffer (b) is copied to the original data address (c), (d). This step is shown at 18. The situation before the attempted write operation is thus restored.

This method is adapted to a multi-stage operation procedure and in practice data will be fed back and forth to the terminal by a serial interface in multiple stages. The sequence register holds information as to the stage in the sequence where interruption takes place. If the original interconnection to the terminal pertains and the operation sequence can be resumed then a re-synchronisation procedure takes place and at 19 there is a check to determine whether copying/re-synchronisation has succeeded. If so then the application program AP is run. If not the software must decide from the state of the sequence register how to re-synchronise the on-card application software and the software communicating with the smart card via the serial line. If data cannot be retrieved from the data copy buffer, and the sequence register indicates that this data should be available, then the smart card is unusable, as indicated at 20.

This may be by virtue of continued failure to implement the application program or positive steps may be taken to invalidate the card as, for example,

by blowing an inbuilt fuse.

The data copy buffer (b) and the data increment buffer (e) must both be large enough to hold the largest possible data block that will be written to EEPROM using this method. An extra 5 bytes of storage are also required (size = 2 bytes, address = 2 bytes, sequence register = 1 byte [at least]). If size can never be greater than 255, then it can be stored in a single byte.

Since the card operates on only one page 8 (Figure 1) at a time in writing, security is enhanced by ensuring that separate EEPROM pages (3 in total) are used for the data copy buffer, the data increment buffer and for the rest of the additional data.

Using this method of writing to EEPROM, the number of bytes actually written to EEPROM is doubled even if a recovery is not invoked (because a copy of the original data must be stored in the data copy buffer before the EEPROM write commences). The total overhead is actually slightly more than this as size, address, and sequence register information must also be written to EEPROM.

Referring now to Figure 3 there is shown the EEPROM configuration for a smart card (otherwise similar to that of Figure 1) to use a METHOD 2 in accordance with the invention. Here respective and separate regions of EEPROM (on respective pages 8) are allocated as:-

(f) a write in progress flag register;

(g) a workspace pointer register;

(h) a size register; and

(i) a data pointer register.

In RAM there is allocated a region (j) as a new data pointer register. Alternatively this may also be in EEPROM.

A flow chart for the writing procedure in METHOD 2 is shown in Figure 4(a). This includes the steps of:-

1. setting a workspace pointer in register (a) to the address of a workspace in EEPROM sufficient in size to hold a contiguous data set corresponding to a size set in register (h) (at 21);
2. copying to the workspace a copy of new data in a region in RAM or EEPROM identified in size by register (h) and in position by register (i) (at 22);
3. setting the write in progress flat (f) (at 23);
4. setting the address in register (i) to the workspace address (at 24); and
5. clearing the write in progress flag in register (f).

The check and recovery procedure for METHOD 2 is shown in Figure 4(b). On reset at 25 the write in progress flag is checked at 26. If cleared the application program AP is run at 27. If not then the last two steps (4 and 5) of the write procedure are repeated. Thus, the data pointer (i) is set equal to the workspace pointer (g) at 28 and the write in progress flag (f) is cleared at 29. If this write procedure succeeds (check at 30) the program AP is executed. If not, then the smart card is unusable (at 31).

If an area of EEPROM is found where an EEPROM

write cannot be completed, then this method readily allows the smart card application software to mark this area as unusable (permanently), and choose another area for data storage. This can greatly extend the life of the smart card (which will very probably be limited by the maximum possible number of EEPROM writes that the smart card is capable of performing), however this is at the expense of maintaining a pointer (a 2 byte overhead) to each data structure stored in EEPROM.

Under normal conditions, the Write in Progress flag is only set for the time required to update a pointer in EEPROM. This is the minimum possible theoretical update time, which should help to ensure that the recovery mechanism is invoked only very rarely. This minimises the number of attempted writes to EEPROM, and thus extends the life of the smart card.

Each data structure written to EEPROM using this method will be extended by two bytes, as a pointer to the data must be continuously maintained. There is a small overhead on each EEPROM read as all data which uses this method must be accessed via a pointer.

The EEPROM pointed to by the Workspace Pointer must be large enough to hold the largest possible data structure that will be written to EEPROM using this method. This space is only required until the EEPROM write has been successfully completed, at which point an equivalent length of EEPROM storage (which used to contain the original data) is released. An extra 7 bytes of storage are also required (Write in Progress flag = 1 byte, New Data Pointer = 2 bytes, Workspace Pointer = 2 bytes, Size = 2 bytes). If Size can never be greater than 255, then it can be stored in a single byte.

Using this method of writing to EEPROM, the data structure is only written to EEPROM once, but three pointers have to be updated (the New Data Pointer, the Workspace Pointer and the Data Pointer - in that order). The Size, Address and Sequence Register information must also be written to EEPROM.

Referring now to Figure 5 there is shown EEPROM allocation for a METHOD 3 of implementing the invention. It is to be understood that the EEPROM of Figure 5 is incorporated in a smart card otherwise similar to that of Figure 1. In Figure 5, separate regions of EEPROM (on separate pages 8) are allocated as:-

(k) a state flag register;
(l) a size register;
(m) an address register; and
(n) an update copy buffer.

The writing procedure in METHOD 3 is illustrated in Figure 6(a). The following steps are implemented:-

1. Copy new data into buffer (n) (at 32);
2. Set state flag (k) (at 33);
3. Copy the new data to EEPROM region identified by size (l) and address (m) (at 34); and
4. Clear state flag (k) (at 35).

The check and recovery procedure illustrated in Figure 6(b) has reset at 36, and a check for the setting of state flag (k) at 37. If the flag is not set then application program AP is run at 38. Otherwise the new data residing in buffer (n) is copied to the region (l), (m) at 39 and the state flag (k) is cleared at 40. If successful, the application program is run. If not, the card is useless (41).

An additional Data area (buffer n) must be large enough to store the largest amount of data which will be written to EEPROM, plus 5 bytes (Size = 2 bytes, Address = 2 bytes, State Flag - 1 byte). If Size can never be greater than 255, then it can be stored in a single byte.

Using this method of writing to EEPROM, the number of bytes actually written to EEPROM is doubled even if a Recovery is not invoked (because a copy of the data must be written to EEPROM). The total overhead is actually slightly more than this as Size, Address must also be written to EEPROM.

To be able to tell that data has not been altered, error detection techniques must be implemented. Error detection usually comprises calculating a checksum whenever the data is updated, storing this checksum, and verifying that it is correct during every subsequent data read. The actual method used to calculate the error detection checksum is irrelevant for the purposes of this document, indeed some smart cards have error detection processes built into the EEPROM hardware, and their particular method of operation may well not be known.

An EEPROM write is deemed to be complete only when the error detection system has been appropriately updated, and has been verified correctly.

Each byte of EEPROM can only be changed a finite number of times before it ceases to function correctly. This is typically 10^5 to 10^6 write cycles. Therefore there is a finite chance of data being altered whilst it resides in EEPROM, and an EEPROM read must only be accepted as valid if the error detection system verifies that the data has not been altered. If an error is detected during an EEPROM read, it probably means that one or more bytes in the

smart card's EEPROM have reached the end of their active life.

Using one of the methods of writing to EEPROM described above ensures that error correction (as opposed to error detection) is not required. Either the EEPROM operation takes place successfully, or the smart card is unusable. There are no circumstances in which an error needs to be corrected. This simplifies the software and reduces the data storage requirements, as error correction is computationally intensive and requires more dedicated bytes of storage than error detection.

One of the three methods of writing data to EEPROM described above (Method Number 1) explicitly keeps a counter (Sequence Register) which stores knowledge of the last successful operation in the series of operations performed during writing to EEPROM. Methods 2 and 3 may have, but do not explicitly require a counter of this type as they reply upon flags which hold information showing whether or not writing to EEPROM has successfully completed.

Even though a method of writing to EEPROM does not always explicitly require a numeric counter, it should be clearly noted that in many systems it will be necessary to maintain such a counter so that interrupted processes of any kind can be restarted. It is of course vitally important for this counter to be written to EEPROM in a secure manner, as if it is not correct it cannot be relied upon by smart card application software attempting to restart an interrupted process.

CLAIMS

1. A method of writing data to non-volatile memory in an integrated circuit device, the device having an interface for temporary connection to a terminal unit; a microprocessor; random access memory and non-volatile memory, the method consisting in allocating a first region of the non-volatile memory for data to be written, allocating a second region of non-volatile memory for write status information, performing a data write operation to write data to said first region, and writing information to said second region signifying a valid data write if, and only if, the data write operation is performed satisfactorily.

2. A method of writing data to non-volatile memory as claimed in Claim 1 wherein the non-volatile memory is divided into pages and write operations are performed on only one page at a time, the said first and second regions of memory being on different pages.

3. A method of writing data to non-volatile memory as claimed in Claim 1 or Claim 2 wherein the non-volatile memory is electrically erasable programmable read-only memory (EEPROM).

4. A method of utilisation of an integrated circuit device to which data has been written in accordance with the method of any of Claims 1 to 3, the device including in the non-volatile memory an application program which controls the microprocessor to run a particular application under normal circumstances, the utilisation method including the step of initially reading the said second portion of the non-volatile memory to derive write status information therefrom and, if the write status information indicates an incomplete write operation, by-passing said application program.

5. A method of utilisation of an integrated circuit device as claimed in Claim 4 including on detection of an incomplete write operation, a procedure of data recovery effective to restore the device to a condition in which the last data write is correct and the status information in the second region of memory reflects this.

6. A method of utilisation of an integrated circuit device as claimed in Claim 5 which includes the step of monitoring the procedure of data recovery and rendering running of the application program impossible if the data recovery procedure fails.

7. A method of utilisation of an integrated circuit device as claimed in Claim 6 which includes the step of permanently disabling the device if the data recovery step fails.

8. A method of utilisation of an integrated circuit device as claimed in any of Claims 5 to 7 wherein said second region of memory is a status register, said status information is indicative of the last satisfactorily performed stage of a multi-stage operation sequence and said data recovery procedure is effective to recover the multi-stage operation sequence from the stage at which it failed, as indicated by the status register.

9. A method of utilisation of an integrated circuit device as claimed in Claim 8 wherein respective and separate regions of the non-volatile memory are allocated as:-

(a) a sequence register which is said second region of memory;

(b) a data copy buffer;

(c) a size register; and

(d) an address register

and allocating a region of RAM or non-volatile

memory as (e) a data incremental buffer, the said first region of non-volatile memory being identified in size and address by data written in memory regions (c) and (d), the said method of writing consisting in:-

1. ensuring that the buffer (e) contains a valid data increment;
2. placing a copy of data to be updated in the buffer (b);
3. incrementing the register (a);
4. incrementing the data at the first region of memory by the amount in buffer (e) and writing the incremental amount to the first region of memory; and
5. incrementing the sequence register (a).

10. A method of utilisation of an integrated circuit device as claimed in Claim 9 wherein the recovery procedure includes copying the data from the data buffer (b) to said first region of memory.

11. A method of utilisation of an integrated circuit device as claimed in any of Claims 5 to 7 wherein said second region of memory is a flag region and said status information is a flag which is set if the said write operation is verified as satisfactory and which is otherwise not set.

12. A method of utilisation of an integrated circuit device as claimed in Claim 11 wherein respective and separate regions of the non-volatile memory are allocated as:-

(f) a write in progress flag register, which is said second region of memory;
(g) a workspace pointer register;
(h) a size register; and
(i) a data pointer register

and allocating a region of RAM or non-volatile

memory as (j) a new data pointer register, the said first region of non-volatile memory being identified in size and position by data written in memory regions (g) and (h), the said method of writing consisting in:-

1. setting a workspace pointer in register (g) to the address of non-volatile memory workspace sufficient to hold a contiguous data set corresponding to a size set in register (h);
2. copying to the workspace a copy of new data identified in address by the new data pointer at (j) and in size by the size data at (h);
3. setting the write in progress flag at (f);
4. setting an address in data pointer register (i) to the address of the workspace; and
5. clearing the write in progress flag in register (f).

13. A method of utilisation of an integrated circuit device as claimed in Claim 12 wherein the recovery procedure comprises the steps of setting the address in data pointer register (i) to the address of the workspace and clearing the write in progress flag in register (f).

14. A method of utilisation of an integrated circuit device as claimed in Claim 11 wherein respective and separate regions of the non-volatile memory are allocated as:-

(k) a state flag register which is said second region of memory;

(l) a size register;

(m) an address register; and

(n) an update copy buffer

the said first region of non-volatile memory being identified in size and position by data written in registers (l) and (m), the said method of writing consisting in:-

1. copying new data to be written into buffer (n);

2. setting the state flat in register (k);

3. writing said new data to be written to said first region of non-volatile memory; and

4. clearing the state flag in register (k).

15. A method of utilisation of an integrated circuit device as claimed in Claim 14 wherein the recovery procedure comprises the steps of copying the contents of the update copy buffer (n) to the said first region of non-volatile memory identified by the contents of registers (l) and (m) and clearing the flag in register (k).

16. An integrated circuit device having an interface for temporary connection to a terminal unit; a microprocessor; random access memory and non-volatile memory, the non-volatile memory including a program for controlling the microprocessor to effect any of the data writing or utilisation methods as claimed in any of the preceding claims.

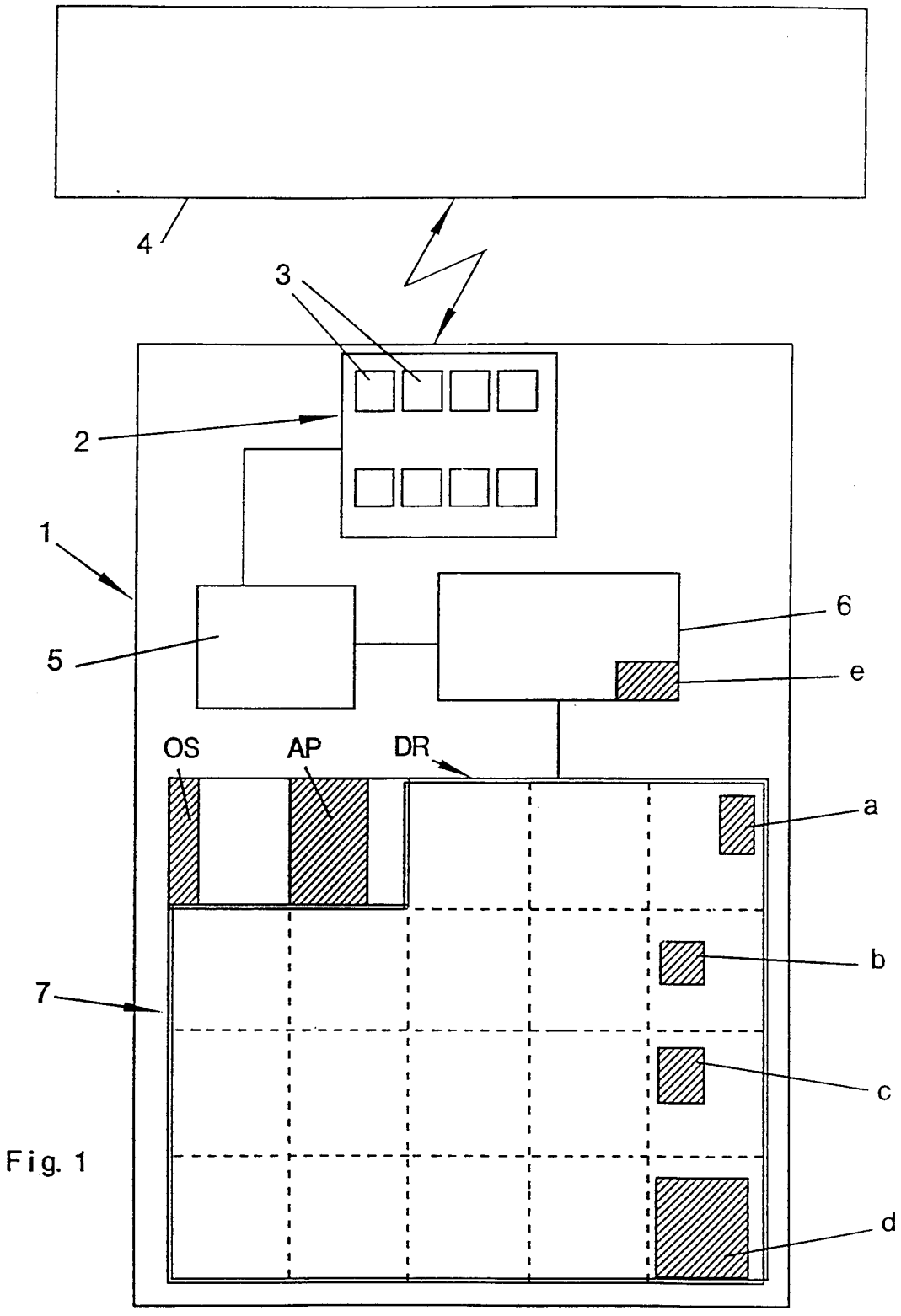


Fig. 1

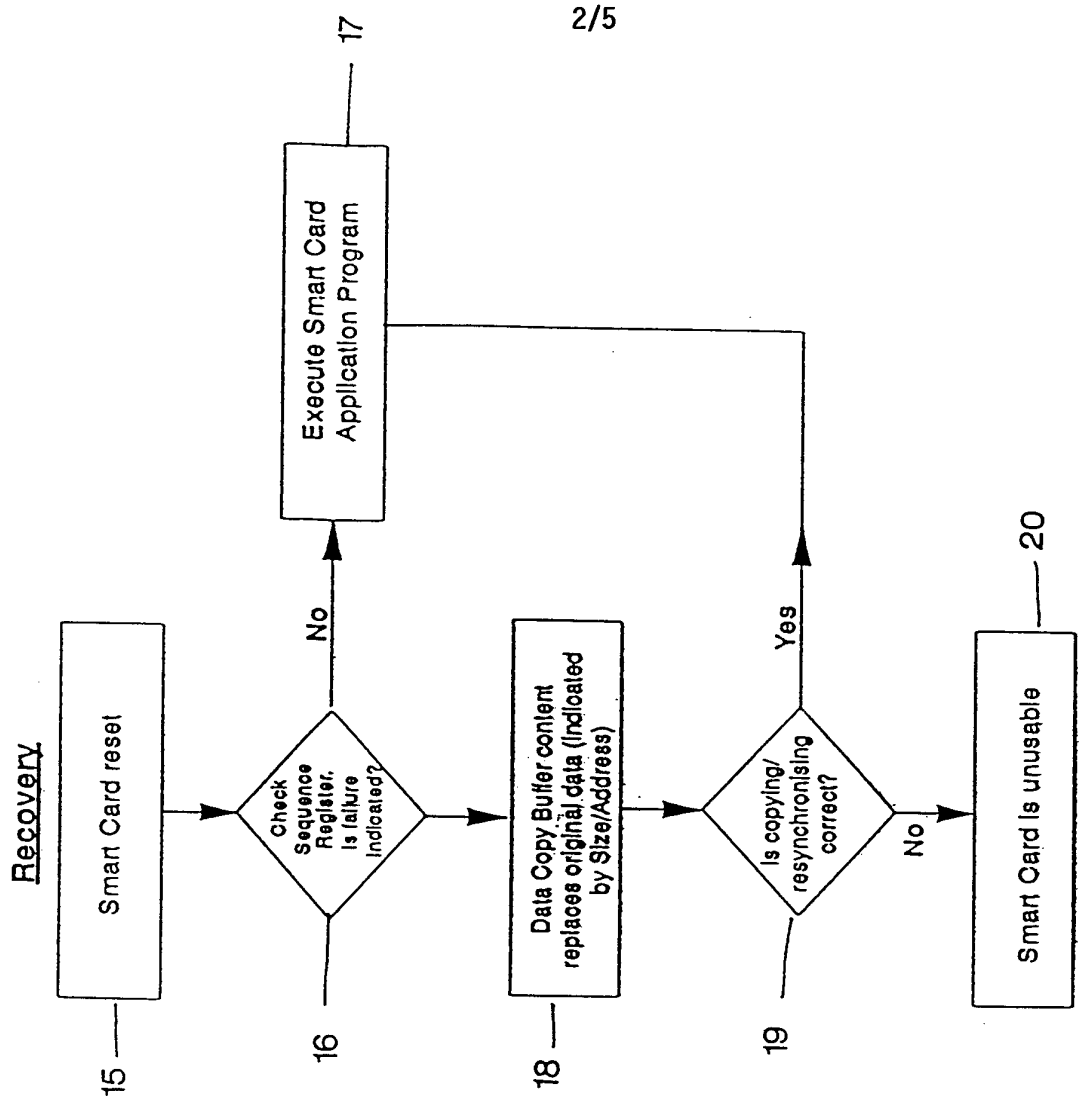

Recovery
Smart Card reset
15
Check Sequence Register. Is failure indicated?
16
No
Execute Smart Card Application Program
17
Data Copy Buffer content replaces original data (indicated by Size/Address)
18
Is copying/ resynchronising correct?
19
Yes
No
Smart Card is unusable
20

Fig. 2(b)

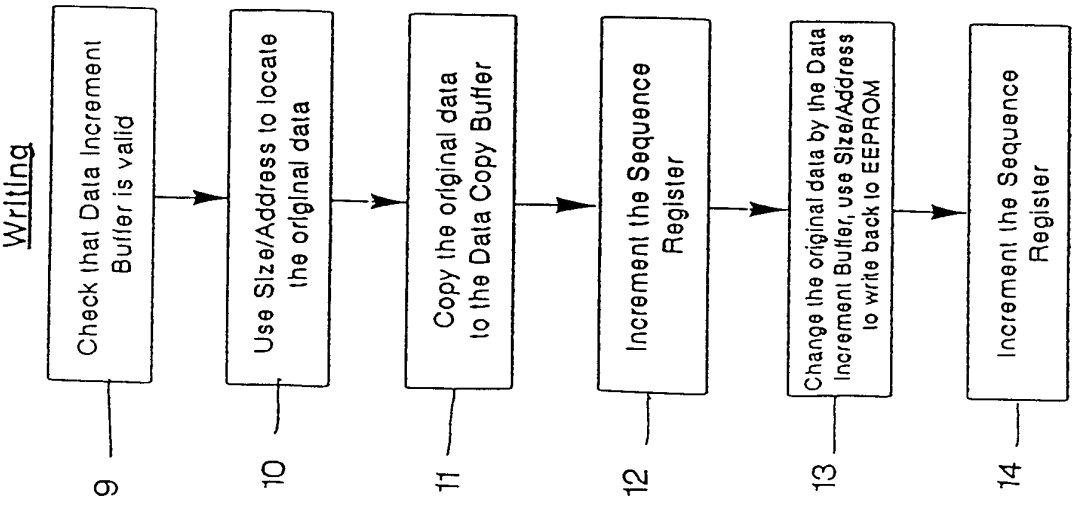

Writing
Check that Data Increment Buffer is valid
9
Use Size/Address to locate the original data
10
Copy the original data to the Data Copy Buffer
11
Increment the Sequence Register
12
Change the original data by the Data Increment Buffer, use Size/Address to write back to EEPROM
13
Increment the Sequence Register
14

Fig. 2 (a)

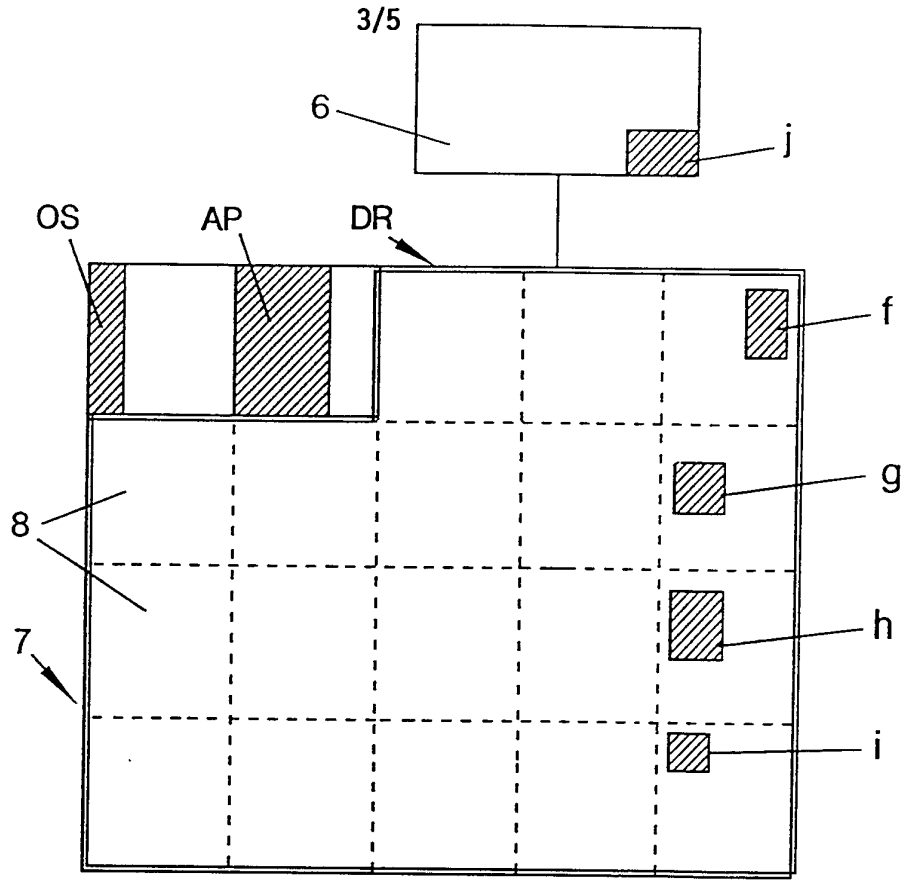
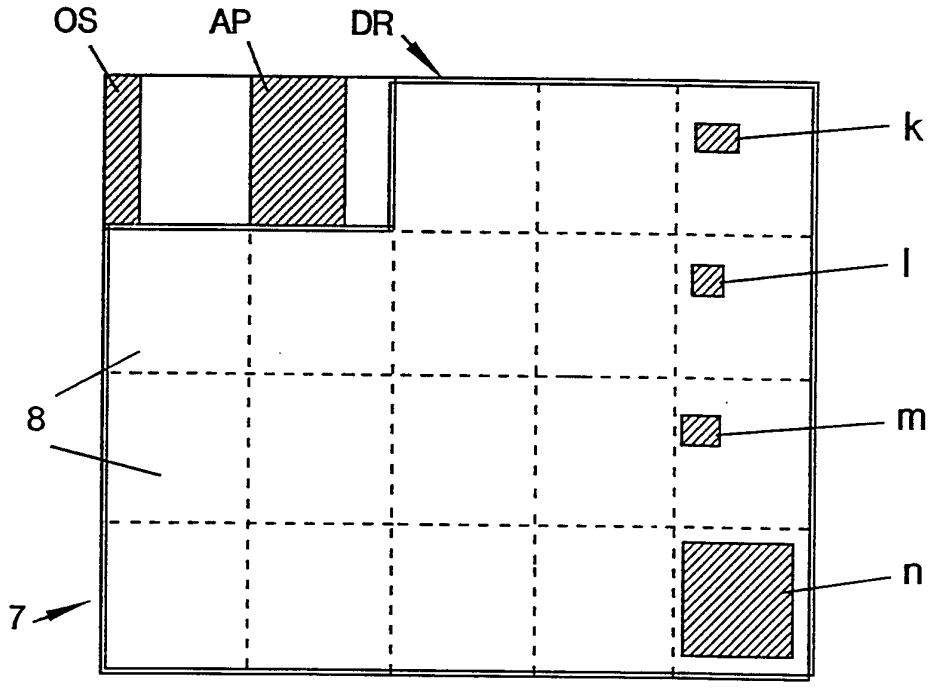


Fig. 3



Fig, 5

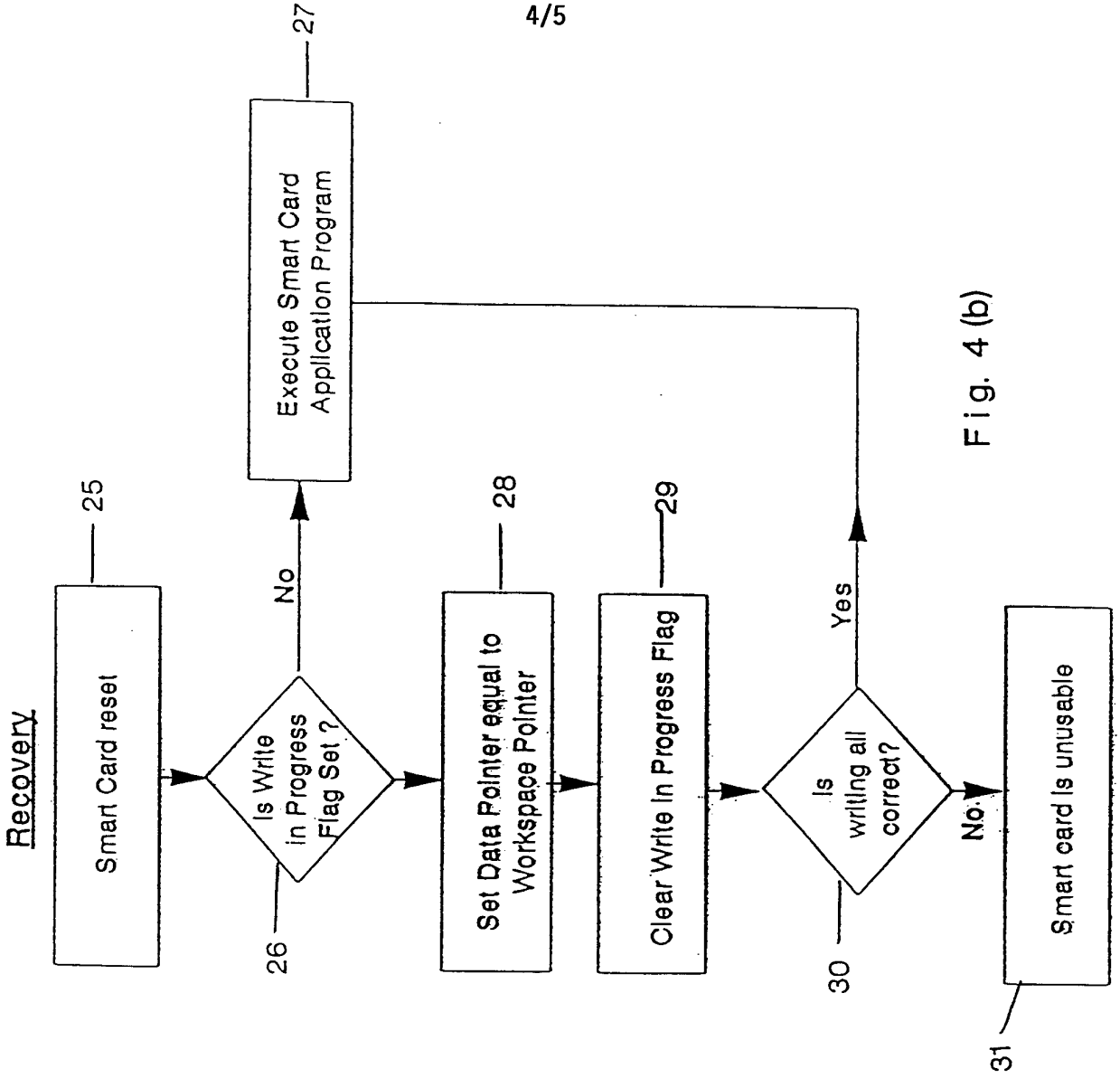

Recovery
Smart Card reset
25
Is Write in Progress Flag Set ?
26
No
Execute Smart Card Application Program
27
Set Data Pointer equal to Workspace Pointer
28
Clear Write In Progress Flag
29
Is writing all correct?
30
Yes
No
Smart card is unusable
31

Fig. 4 (b)

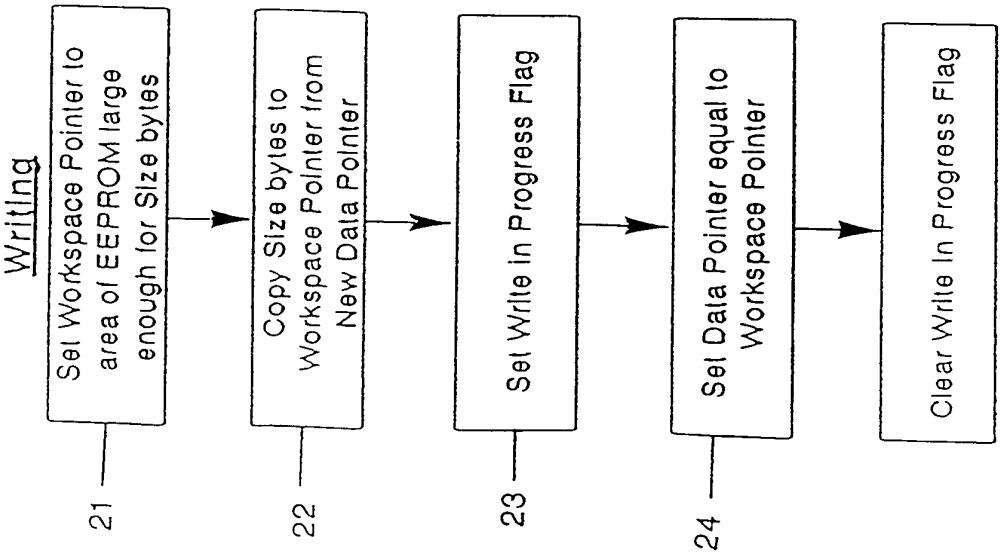

Writing
Set Workspace Pointer to area of EEPROM large enough for Size bytes
21
Copy Size bytes to Workspace Pointer from New Data Pointer
22
Set Write In Progress Flag
23
Set Data Pointer equal to Workspace Pointer
24
Clear Write In Progress Flag

Fig. 4 (a)

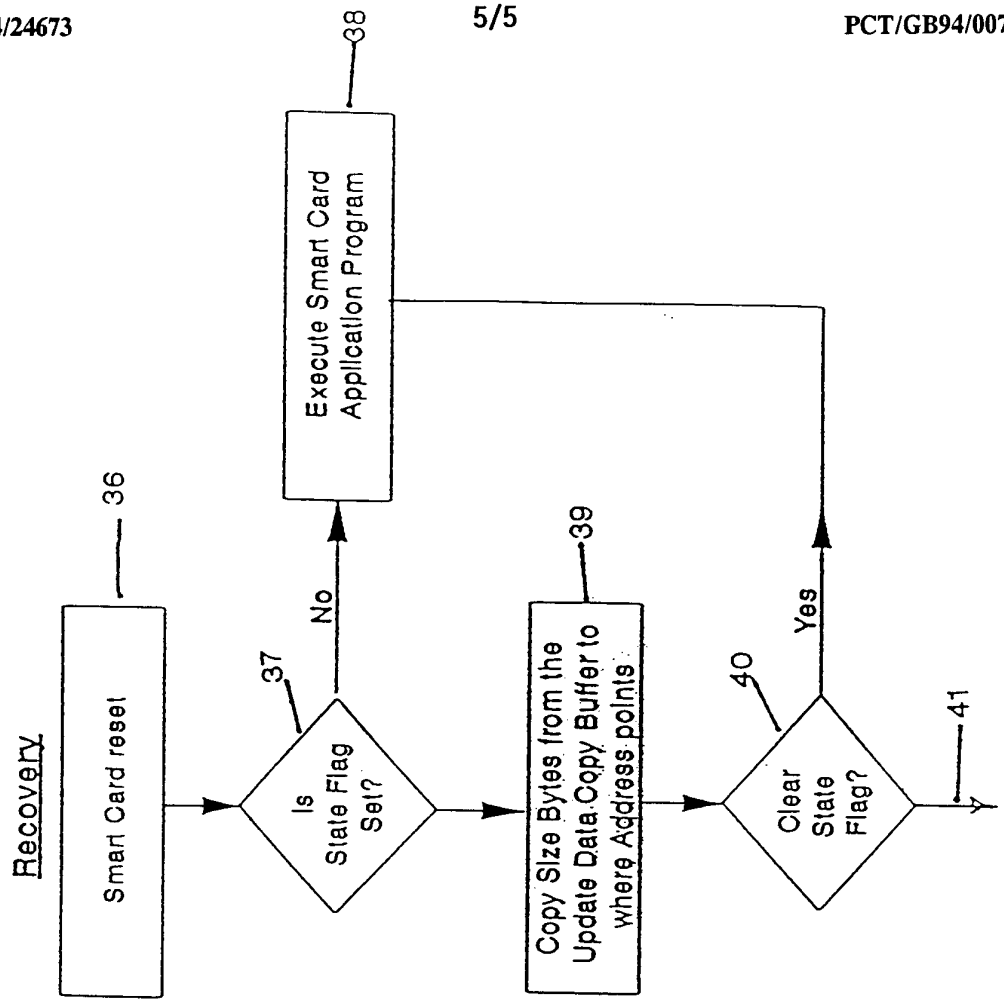

Recovery
Smart Card reset
36
Is State Flag Set?
37
No
Execute Smart Card Application Program
38
Copy Size Bytes from the Update Data.Copy Buffer to where Address points
39
Clear State Flag?
40
Yes
41

Fig. 6 (b)

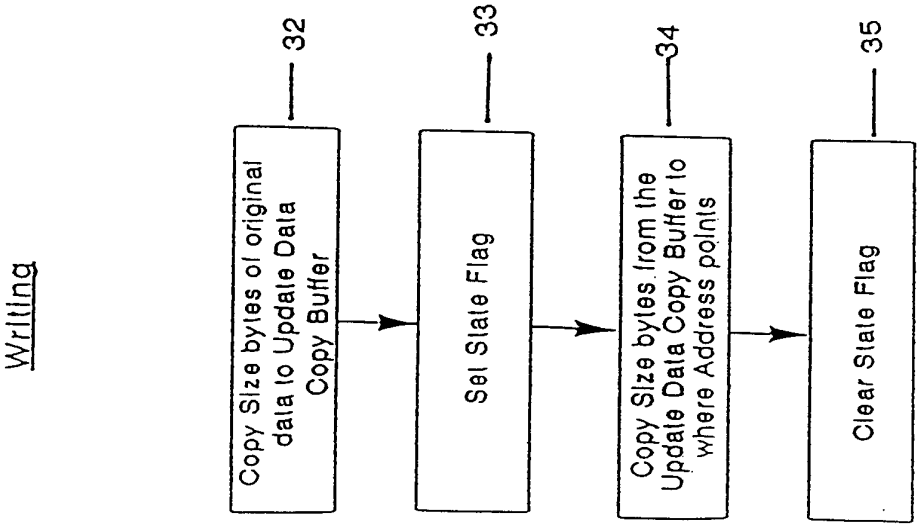

Writing
Copy Size bytes of original data to Update Data Copy Buffer
32
Set State Flag
33
Copy Size bytes from the Update Data Copy Buffer to where Address points
34
Clear State Flag
35

Fig. 6 (a)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/GB 94/00775

A. CLASSIFICATION OF SUBJECT MATTER
IPC 5 G11C16/06 G06F11/14

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)
IPC 5 G11C G06F

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 practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X A	WO,A,89 10618 (SCIENTIFIC ATLANTA INC.) 2 November 1989 see abstract; see page 6, line 32 - page 9, line 5; claim 2	1,16 2-5, 8-11,14, 15
X A	WO,A,92 04716 (GEMPLUS CARD INTERNATIONAL) 19 March 1992 see page 3, line 13 - page 14, line 18; figures 1-4	1,3,16 2,4-6, 8-11,14, 15
X A	EP,A,0 026 980 (FUJITSU FANUC LIMITED) 15 April 1981 see page 7, line 15 - page 17, line 1; figures 2,3	1,2 4,5, 8-11,14, 15

-/--

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

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

"E" earlier document but 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
12 July 1994	20.07.94
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax (+31-70) 340-3016	**Authorized officer** Cummings, A

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

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP,A,0 398 545 (DELCO ELECTRONICS CORPORATION) 22 November 1990 see column 3, line 1 - column 5, line 8; figure 1 ---	1-3
A	IEEE DIGEST INT. SYMPOSIUM ON FAULT TOLERANT COMPUTING, 20 June 1973, NEW YORK, US pages 11 - 16 ROHR, JOHN A 'STAREX self-repair routines: software recovery in the JPL-STAR computer' -----	

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/GB 94/00775

Patent document cited in search report	Publication date	Patent family member(s)		Publication date
WO-A-8910618	02-11-89	US-A-	4922456	01-05-90
		AU-A-	3560489	24-11-89
WO-A-9204716	19-03-92	FR-A-	2666425	06-03-92
		EP-A-	0546048	16-06-93
EP-A-0026980	15-04-81	JP-C-	1513282	24-08-89
		JP-A-	56037883	11-04-81
		JP-B-	63053636	25-10-88
		US-A-	4517663	14-05-85
EP-A-0398545	22-11-90	JP-A-	3019053	28-01-91