



(19) **United States**

(12) **Patent Application Publication**  
**Mototani et al.**

(10) **Pub. No.: US 2008/0096672 A1**

(43) **Pub. Date: Apr. 24, 2008**

(54) **GAME DEVICE, AND FRAUD DETECTING METHOD FOR THE GAME DEVICE**

**Publication Classification**

(76) Inventors: **Shigeru Mototani**, Tokyo (JP);  
**Shinichi Furuhashi**, Tokyo (JP);  
**Yusuke Shimizu**, Tokyo (JP)

(51) **Int. Cl.**  
*A63F 9/24* (2006.01)  
(52) **U.S. Cl.** ..... 463/43

Correspondence Address:  
**JORDAN AND HAMBURG LLP**  
**122 EAST 42ND STREET**  
**SUITE 4000**  
**NEW YORK, NY 10168 (US)**

(57) **ABSTRACT**

A game machine can detect the reasons of restart and switch the maintenance level of player information based on the reasons. The game machine has first and second control units respectively controlling the execution of a game program and continuing incrementing a count value in a progressing process of the game program; first and second nonvolatile memory areas storing the count value incremented by the second control unit as a life count and a count value acquired by the first control unit based on the life count as a copy count, respectively, wherein the life count stored in the first nonvolatile memory area and the copy count stored in the second nonvolatile memory area are compared at restart, and judgment is made that the power is shut OFF due to an irregular action of a player before the restart if the life count and the copy count are not in a predetermined relationship.

(21) Appl. No.: **11/904,994**

(22) Filed: **Sep. 28, 2007**

**Related U.S. Application Data**

(63) Continuation of application No. PCT/JP2006/305990, filed on Mar. 24, 2006.

(30) **Foreign Application Priority Data**

Mar. 31, 2005 (JP) ..... 2005-100375

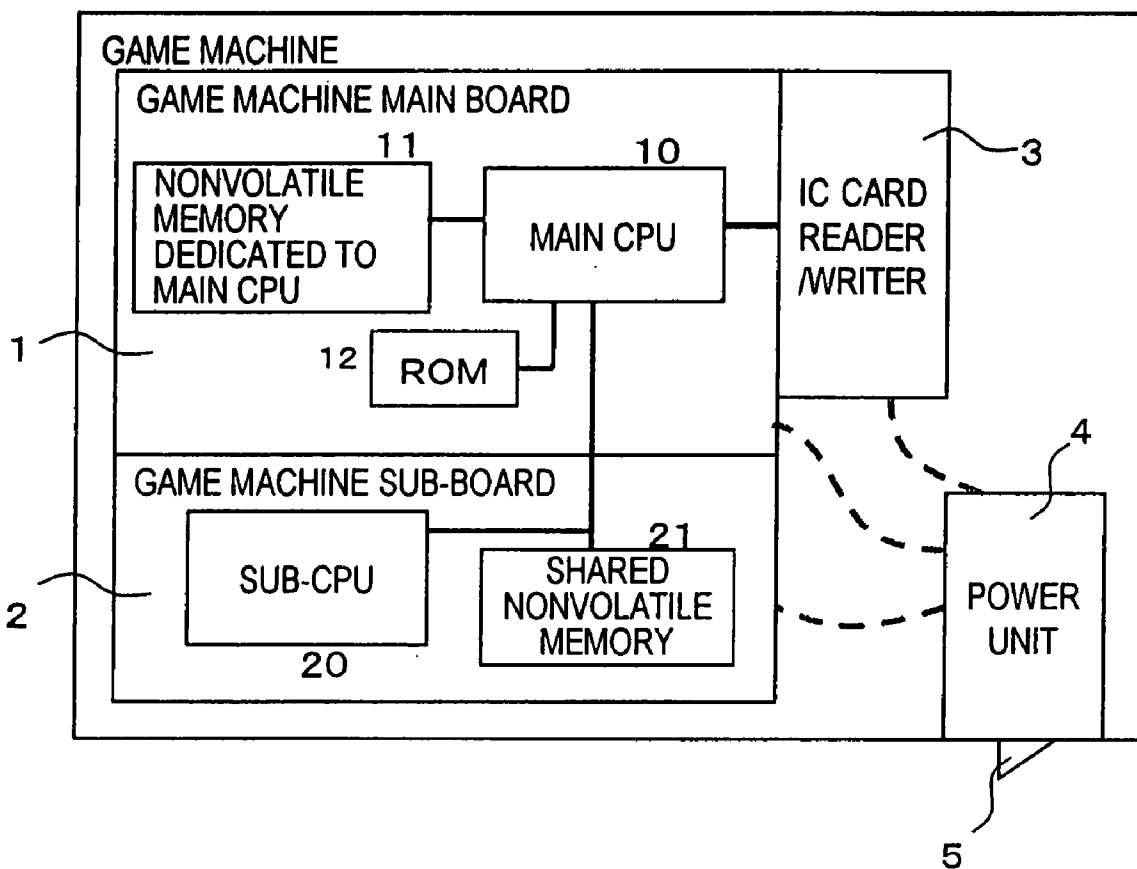


FIG. 1

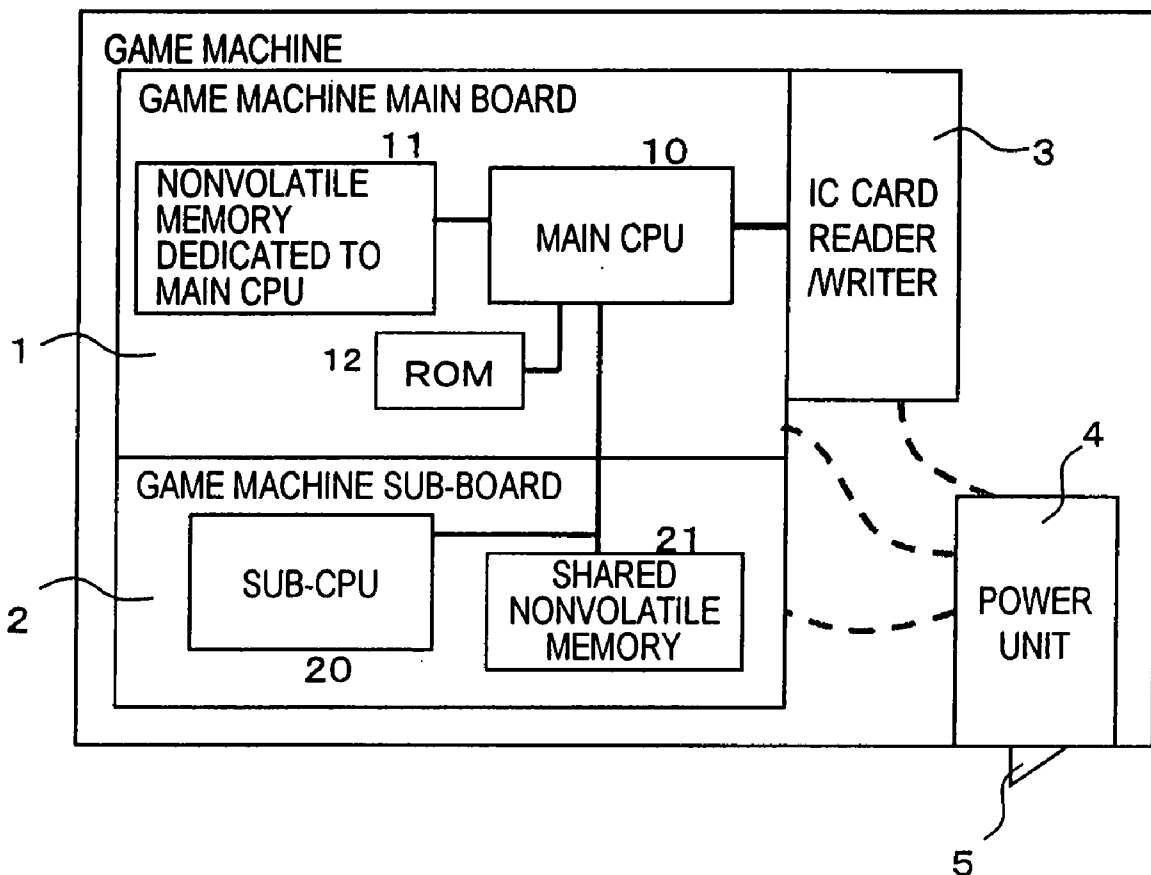


FIG. 2A

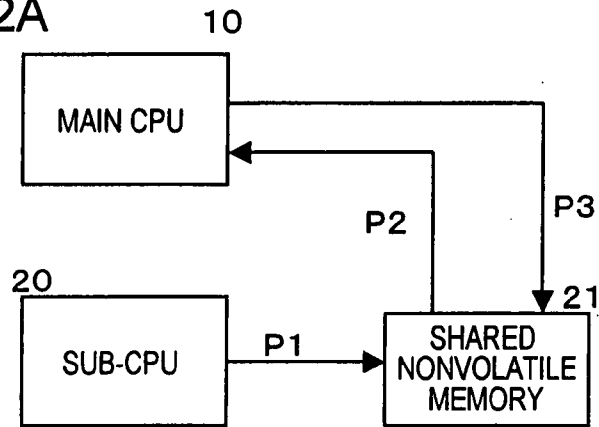


FIG. 2B

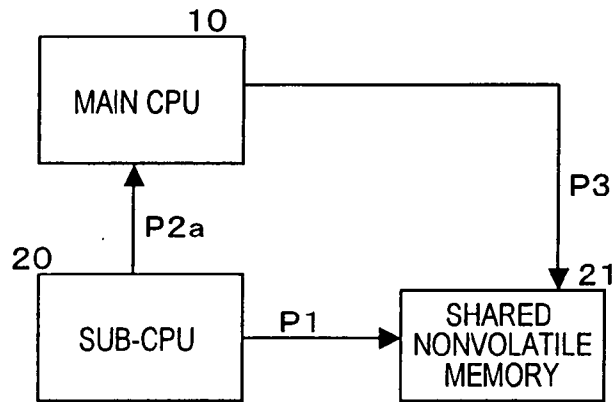


FIG. 2C

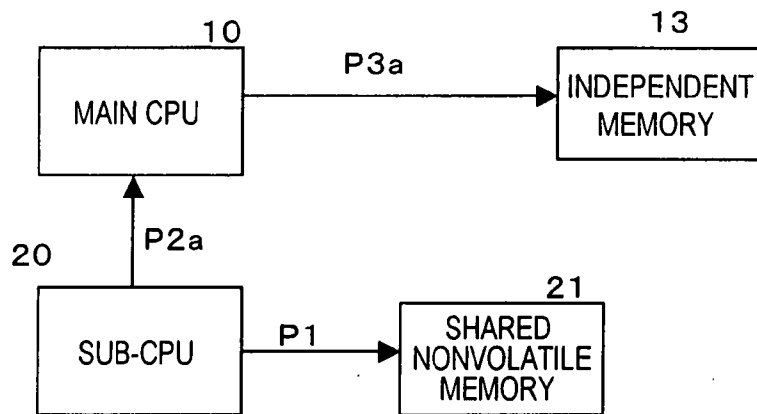


FIG. 3A

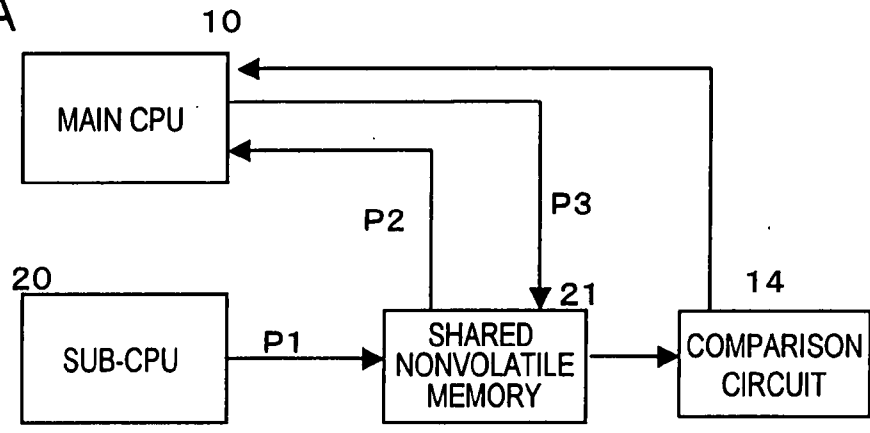


FIG. 3B

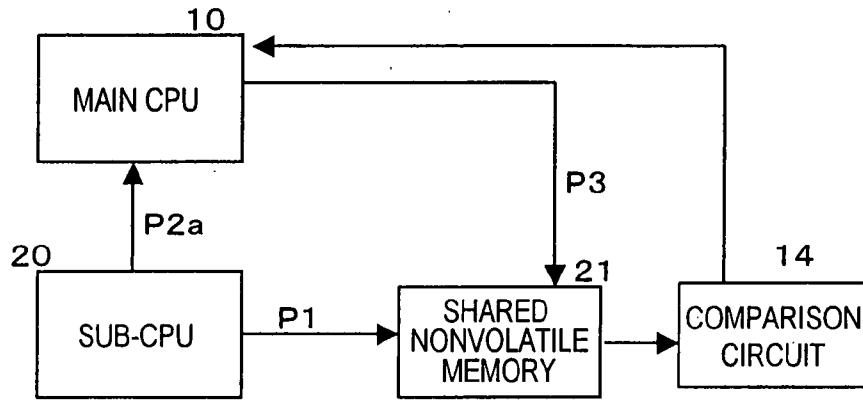


FIG. 3C

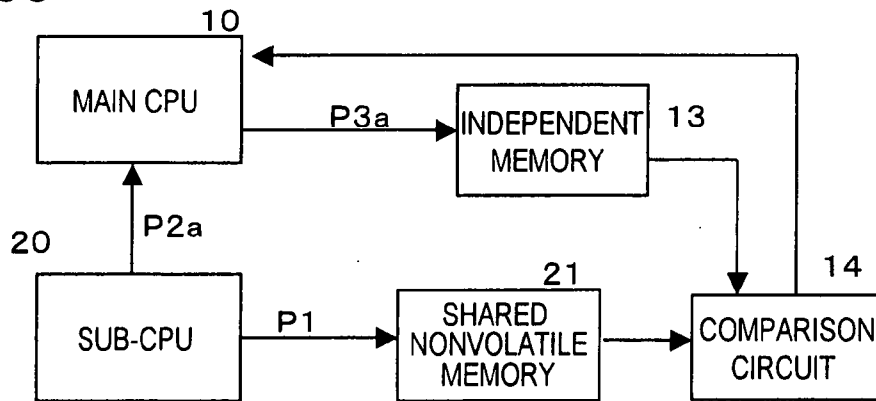


FIG. 4

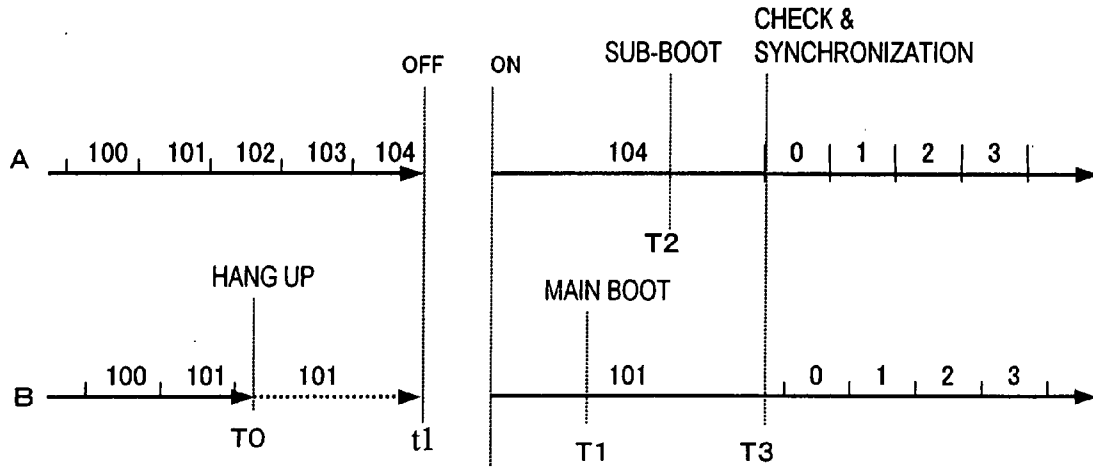


FIG. 5

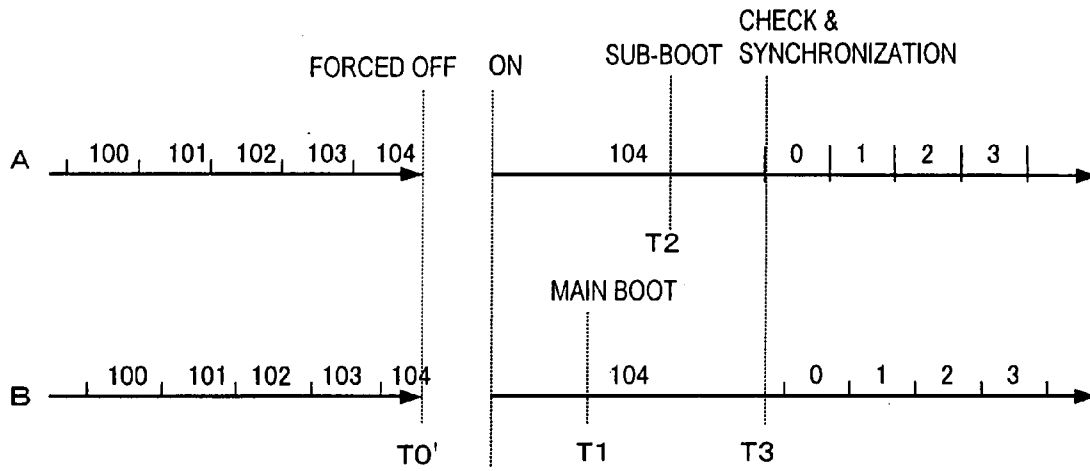


FIG. 6

PREVIOUS STARTUP STATE		COUNT STATE		
MAIN BOARD	SUB-BOARD	LIFE COUNT	COPY COUNT	
I	NORMAL	NORMAL	$N + a$ ( $a \leq 1$ )	N
II	HANG UP	NORMAL	$N + a$ ( $a > 1$ )	N
III	FORCED OFF	FORCED OFF	N	N

FIG. 7

CASE	COUNT 1	COUNT 2	COUNT 3	ORIGINAL VALUE
1	X	$N - 1$	$N - 1$	$N - 1$
2	N	X	$N - 1$	$N - 1$ or N
3	N	N	X	N

FIG. 8

LIFE COUNT 1	
LIFE COUNT 2	MULTIPLEX
LIFE COUNT 3	MULTIPLEX
COPY COUNT 1	
COPY COUNT 2	MULTIPLEX
COPY COUNT 3	MULTIPLEX

FIG. 9

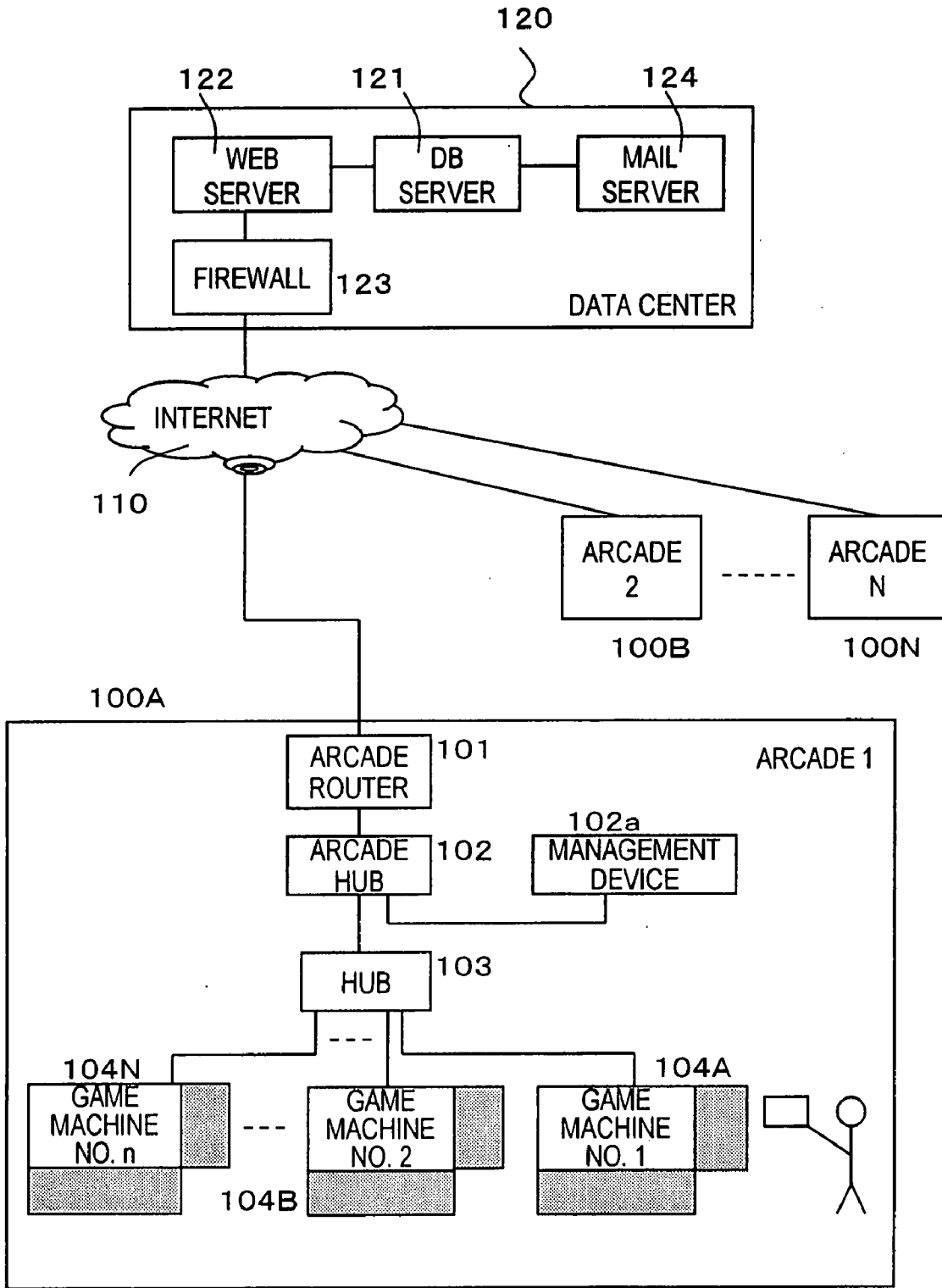




FIG. 10

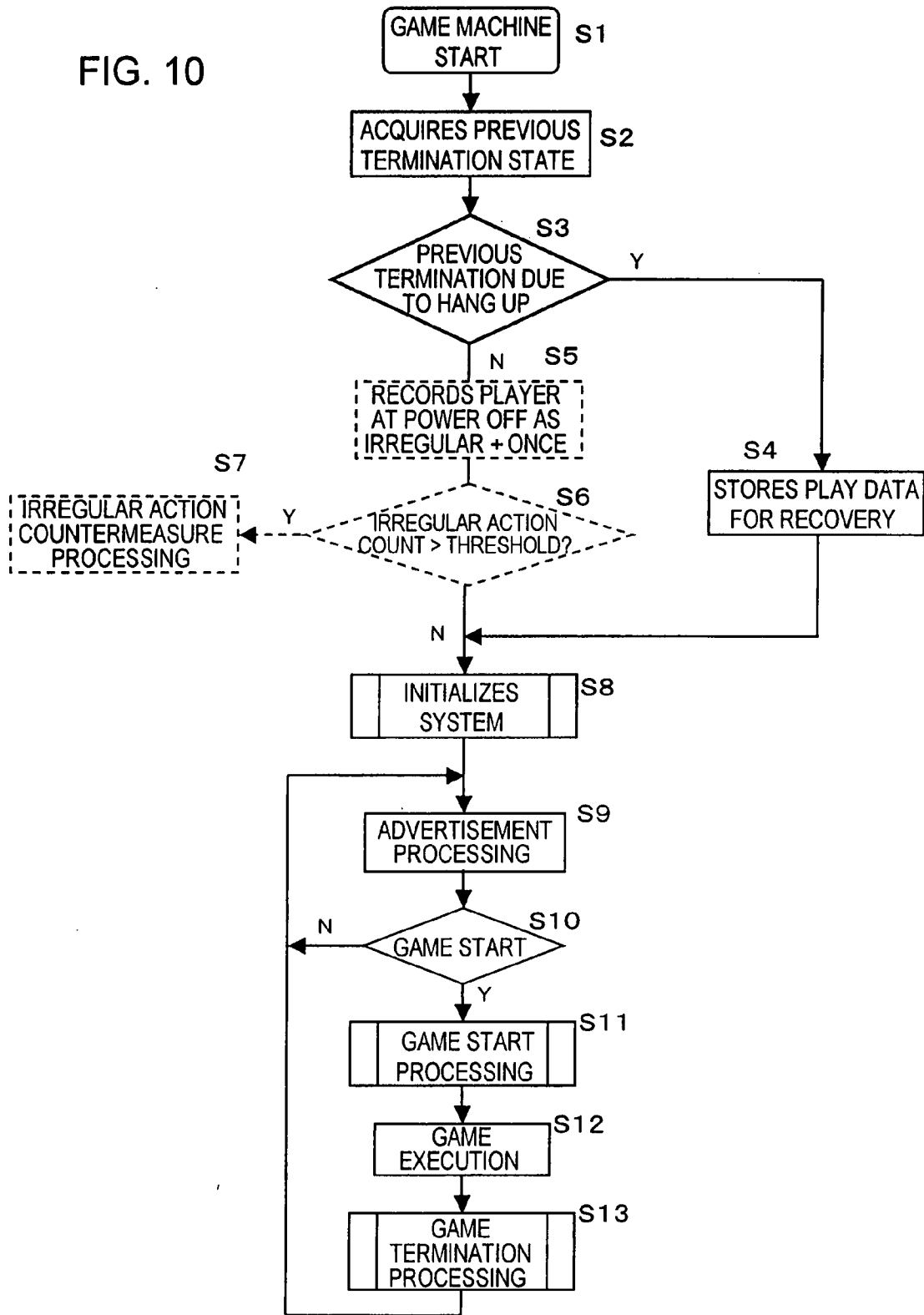


FIG. 11

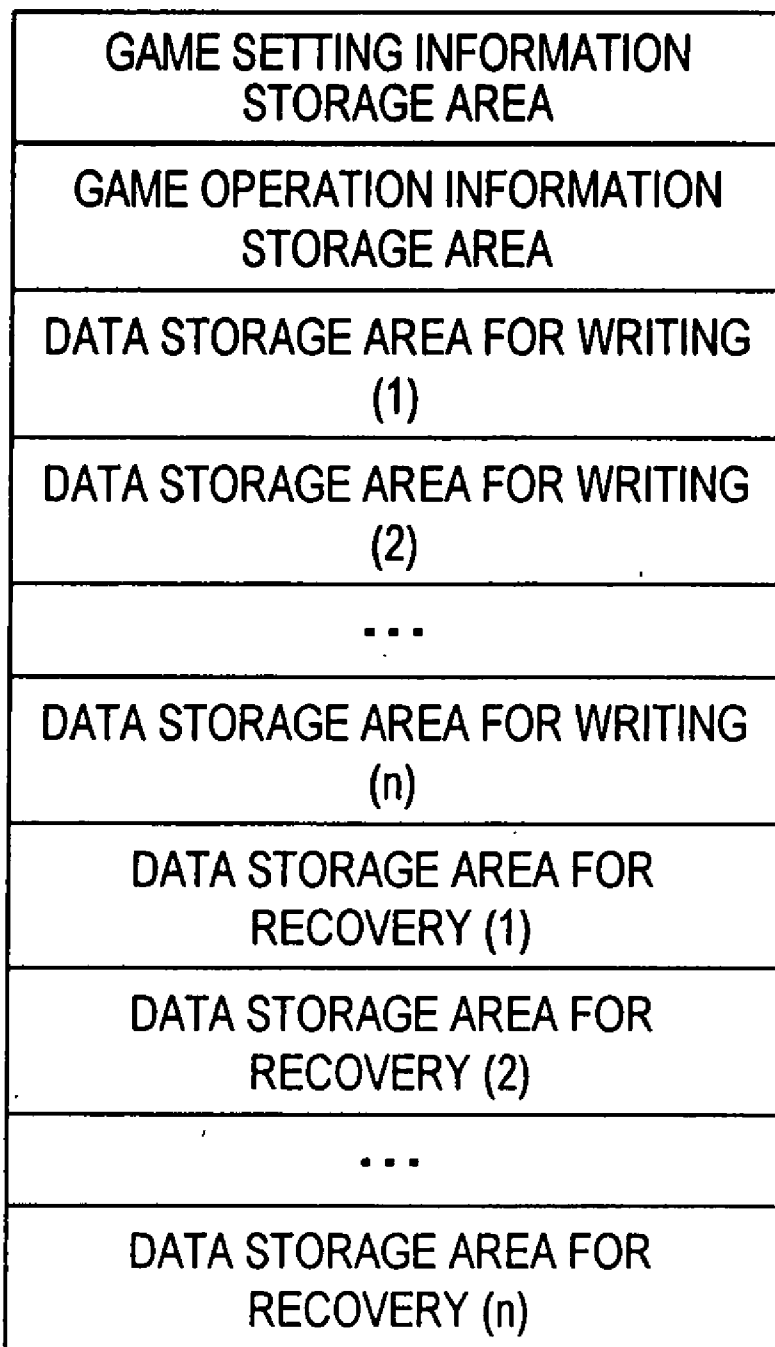


FIG. 12

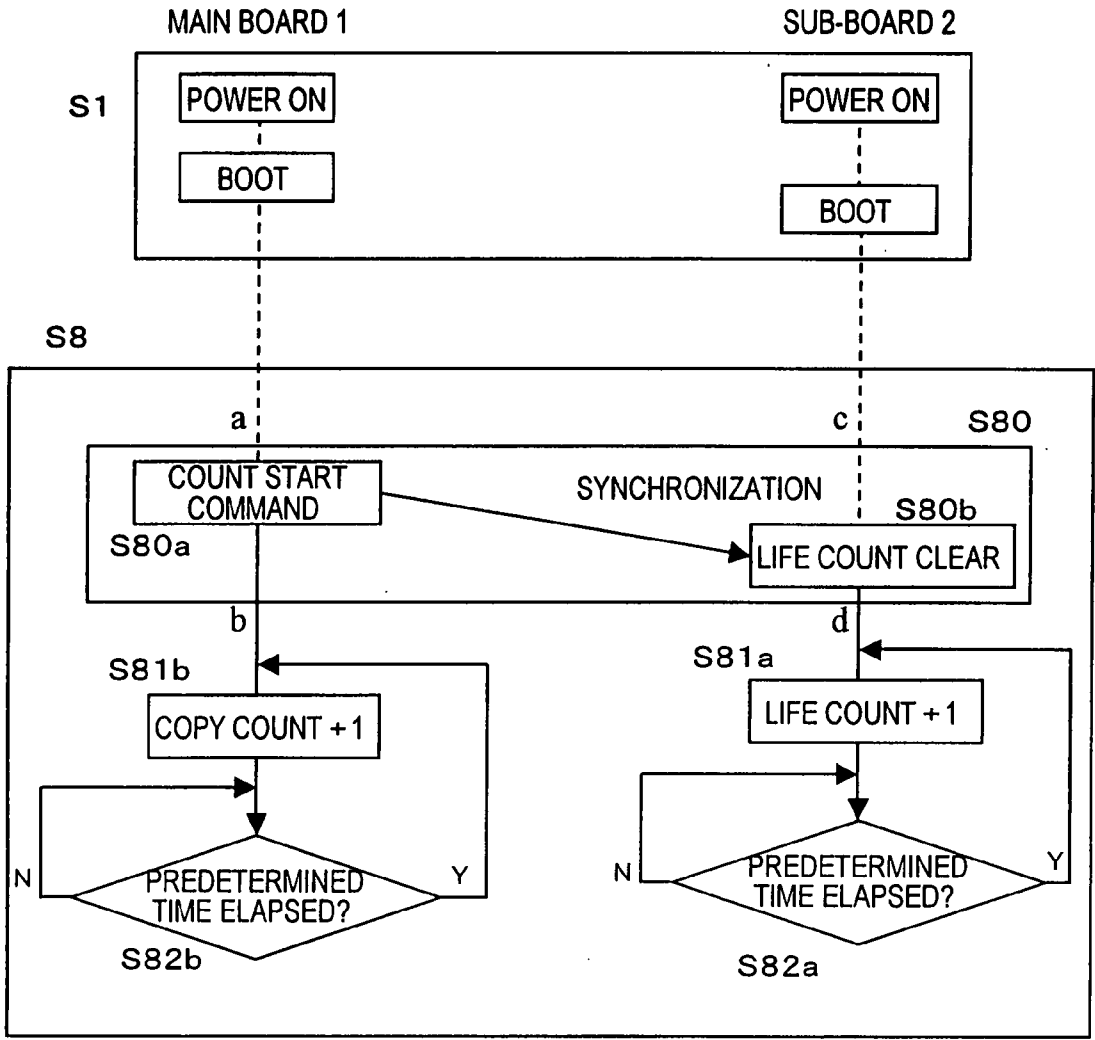


FIG. 13

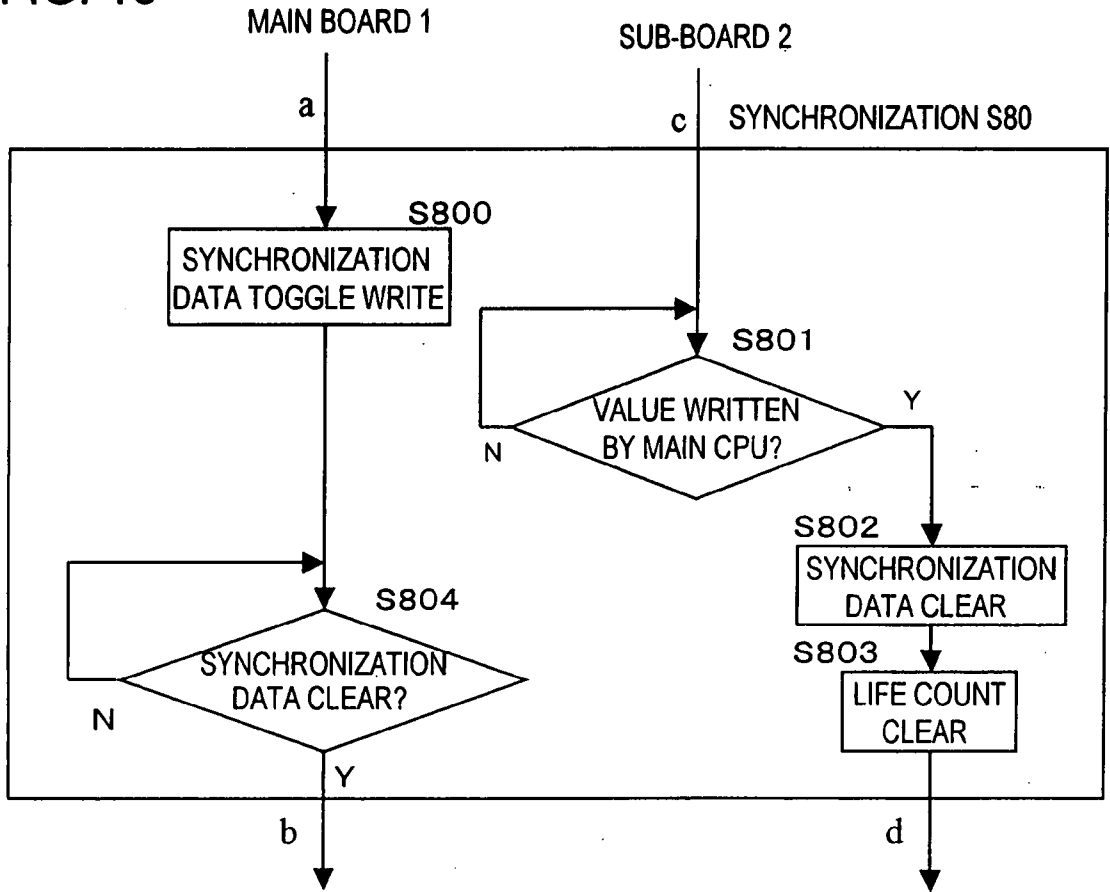
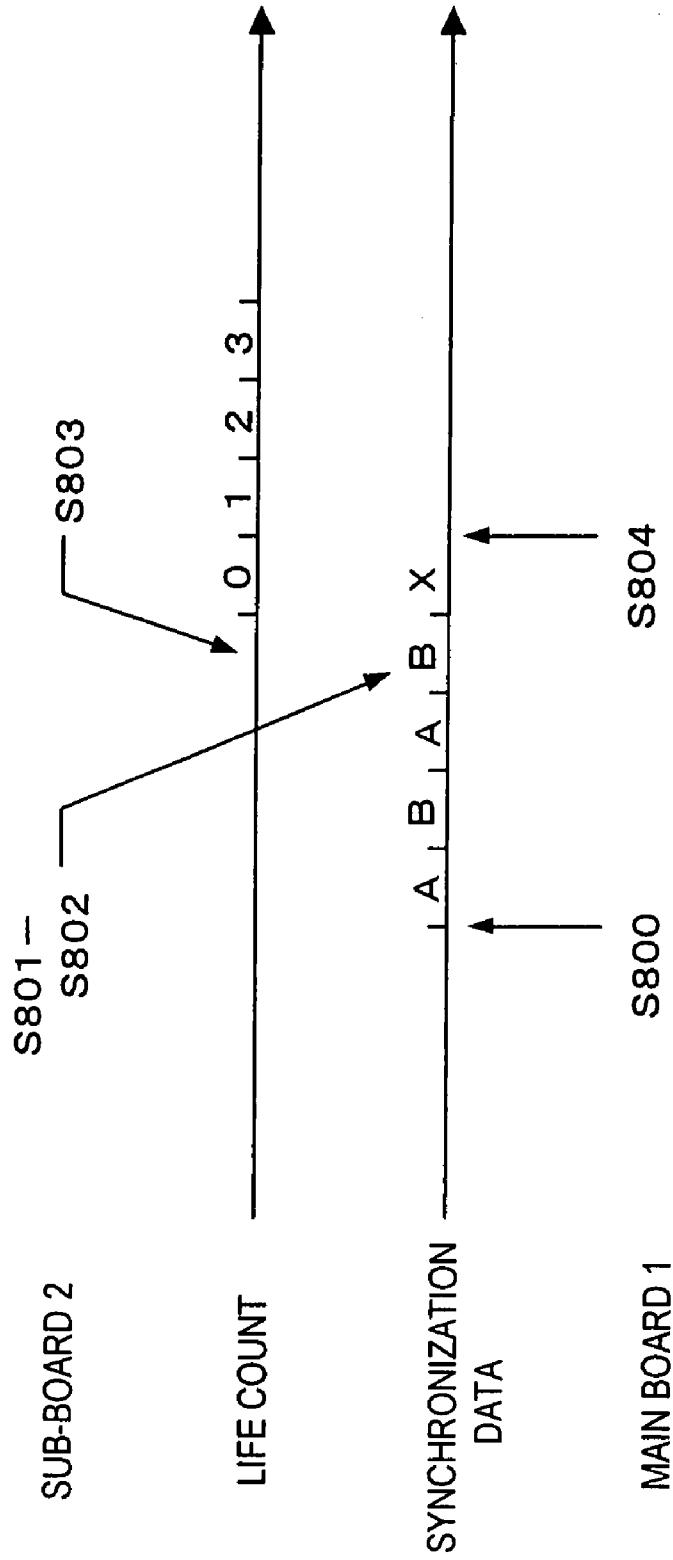


FIG. 14



**GAME DEVICE, AND FRAUD DETECTING METHOD FOR THE GAME DEVICE**

**CROSS REFERENCE TO RELATED APPLICATIONS**

[0001] This application is a continuation of International Application No. PCT/JP2006/305990, filed on Mar. 24, 2006, now pending, herein incorporated by reference. Further, this application is based upon and claims the benefit of priority from the prior Japanese Patent Application No. 2005-100375, filed on Mar. 31, 2005, the entire contents of which are incorporated herein by reference.

**TECHNICAL FIELD**

[0002] The present invention relates to an irregular use detection method for a game machine, and a game machine to which the method is applied.

**BACKGROUND ART**

[0003] Recently the arcade game contents playable for game machines installed in game centers that have a plurality of game machines have become advanced and complicated, and the game machines are constructed so as to have specifications to support these games.

[0004] For example, game machines have a configuration to record information on a player who plays a game (e.g. game achievement record in past) on an IC card or data base (DB) server.

[0005] Because of this, in some cases a player who is serious about their game achievement record may intentionally shut the power of the game machine OFF before the game achievement result is recorded when the loss of a game seems certain.

[0006] On the other hand, programs to be executed by the game machine become complicated as games advance, and unexpected hang-ups may occur. In this case, the store operator managing the game machines turns the power of the game machine OFF once, then ON again to restart it.

[0007] In both cases, continuation of the game execution is disabled once. Therefore it is important to identify whether the cause of discontinuation of the execution of the game is an intentional power disconnection by a player, or a hang up of the system for which the player is not responsible.

[0008] In other words, the cause of disconnection of the power supply when the program is operating normally could be the intention or error of the player, but when an unexpected hang up occurs, the player information must be preserved so that the player does not experience a disadvantage.

[0009] In Japanese Patent Application Laid-Open No. 2001-327712 (see paragraph [0003] and [0096]), referred to as Patent Document 1 here, an invention to prevent the loss of payout information at power failure, which occurred paying out winnings based on winning pachinko balls on a pachinko machine, is disclosed. The invention according to this Patent Document 1 is a pachinko machine having a main side control unit and a sub-side control unit, each of which stores a value (the number of winning pachinko balls paid out) counted during payout in a recording section respec-

tively, and when main power is recovered, information on one of the recording sections of the main side and sub-side is read, and the payout device is driven and controlled based on the payout information before the power failure.

[0010] In Japanese Patent Application Laid-Open No. 2003-271909 (see paragraph [0015]), referred to as Patent Document 2 here, an invention of writing a same count value in three nonvolatile memories, comparing the data of these memories when power is turned ON, and judging the correct data based on a majority decision, is disclosed.

[0011] The inventions according to Patent Documents 1 and 2 are both for acquiring correct information without errors of numeric values due to a power failure, by comparing the information recorded in a plurality of recording sections.

[0012] However, neither documents state a concept of searching for the cause of the disconnection of power of the system, and controlling subsequent processing according to this cause.

[0013] In particular, nothing is stated about the power OFF caused by an irregular action of a player, and how to maintain player information when the game is restarted there after.

**DISCLOSURE OF THE INVENTION**

[0014] With the foregoing in view, it is an object of the present invention to provide an irregular use detection method for a game machine, which can easily detect the cause of restarting the game machine and can switch the maintenance level of the player information based on the reason for the restart, and a game machine system that uses this irregular use detection method.

[0015] A first aspect of the game machine of the present invention to achieve the above object has: a first control unit for controlling the execution of a game program; a second control unit for continuing incrementing a count value in a progressing process of the game program; a first nonvolatile memory area for storing the count value incremented by the second control unit as a life count; and a second nonvolatile memory area for storing a count value acquired by the first control unit based on the life count as a copy count, wherein the life count stored in the first nonvolatile memory area and the copy count stored in the second nonvolatile memory area are compared at restart, and judgment is made that power is shut OFF due to an irregular action of a player before the restart if the life count and the copy count are not in a predetermined relationship.

[0016] A second aspect of the game machine of the present invention to achieve the above objects is the game machine according to the first aspect, further having a managing unit for managing power OFF and a player, wherein when judgment is made that the power is shut OFF due to an irregular action of the player, the management unit controls the player to limit execution of the game.

[0017] A third aspect of the game machine of the present invention to achieve the above object is the game machine according to the first aspect, wherein the copy count to be stored in the second nonvolatile memory area is a copy of the life count.

[0018] A fourth aspect of the game machine of the present invention to achieve the above object is the game machine according to the first aspect, wherein the first nonvolatile memory area and the second nonvolatile memory area are in a common nonvolatile memory.

[0019] A fifth aspect of the game machine of the present invention to achieve the above object is the game machine according to the first or fourth aspect, wherein the first nonvolatile memory area and the second nonvolatile memory area have a multiplexed area respectively, and the same life count and copy count are written in the multiplexed area respectively.

[0020] The sixth aspect of the game machine of the present invention to achieve the above object is the game machine according to the fifth aspect, wherein if the multiplexed area is a triplexed area and if values written in two or more areas of the multiplexed area are the same, then this same value is regarded as the original life count and copy count multiplexed.

[0021] The seventh aspect of the game machine of the present invention to achieve the above object is the game machine according to the sixth aspect, wherein if values written in the two or more areas are not the same, and if a difference of a first value written in a first rank order and a third value written in a third rank order is 1, then the first value or the third value is regarded as the original life count and copy count.

[0022] A first aspect of the game machine system to achieve the above object of the present invention has a plurality of game machines, and a management device for managing power Off and player information of the plurality of game machines, wherein each of the plurality of game machines further has: a first control unit for controlling the execution of a game program; a second control unit to continue incrementing a count value in a progressing process of the game program; a first nonvolatile memory area for storing a count value incremented by the second control unit as a life count; and a second nonvolatile memory area for storing a count value acquired by the first control unit based on the life count as a copy count, wherein the life count stored in the first nonvolatile memory area and the copy count stored in the second nonvolatile memory area are compared at restart, and it is judged that the power was shut OFF due to an irregular action of a player before restart if the life count and the copy count are not in a predetermined relationship, and the management device holds information of the player who committed the irregular action judged above, and controls the player to limit execution of a game on another game machine.

[0023] A second aspect of the game machine system to achieve the above object of the present invention is the game machine system according to the first aspect, wherein the management device further has one data center connected via a network, and the data center further has a data base for holding information on players who execute games, and acquires information on a player who committed an irregular action on a plurality of game machines, which is transmitted via the management device, and registers it in the data base.

[0024] An irregular action judgment method for the game device to achieve the above object of the present invention has steps of: continuing incrementing a count value by a

second control unit in a progressing process of a game program which is executed by a first control unit; storing a count value incremented by the second control unit in a first nonvolatile memory as a life count; acquiring a count value by the first control unit based on the life count; storing the count value acquired by the first control unit in a second nonvolatile memory as a copy count; comparing the life count stored in the first nonvolatile memory area and the copy count stored in the second nonvolatile memory area at restart; and judging that the power was shut OFF due to an irregular action of a player before the restart if the life count and the copy count are not in a predetermined relationship.

[0025] Because of the above characteristics of the present invention, it can easily be judged whether the cause of a restart is an intentional irregular action of a player or an unexpected hang up of the system. As a result, maintenance processing of player information at restart can be more appropriately performed.

[0026] If the cause of a restart is an intentional irregular action of a player, the player information is set so as to be a disadvantage for the player who committed this irregular action, thereby fairness to other players participating in a network game is insured, and enthusiasm to participate in the game is not diminished.

[0027] The characteristics of the present invention will be further clarified by the embodiments of the invention, which will be described below with reference to the drawings.

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0028] FIG. 1 is a diagram depicting a configuration example of a game machine to which the present invention is applied;

[0029] FIG. 2A to 2C are diagrams depicting the relationship of the first memory area and the second memory area;

[0030] FIG. 3A to 3C are diagrams depicting the method for comparing the life count and the copy count;

[0031] FIG. 4 is a diagram depicting the relationship between the life count and the copy count according to the present invention;

[0032] FIG. 5 is a diagram depicting the state of the count value when the power switch 5 is turned OFF by an intentional irregular action of a player in comparison with FIG. 4;

[0033] FIG. 6 is a table to show that a previous startup state can be known by the count state;

[0034] FIG. 7 is a table showing an example when the data to be written in a nonvolatile memory is triplexed;

[0035] FIG. 8 is a table showing areas of the nonvolatile memory corresponding to the triplexing in FIG. 7;

[0036] FIG. 9 is a diagram depicting the configuration example of the game machine system assuming the case when a game is played among game machines in a plurality of arcades via a network;

[0037] FIG. 10 is a diagram depicting the general flow of the game machine system;

[0038] FIG. 11 shows an example of play data recorded in the nonvolatile memory 11 dedicated to the main CPU 10;

[0039] FIG. 12 is a diagram depicting one method for initializing the system;

[0040] FIG. 13 is a diagram depicting a flow of another method of synchronization processing (step S80) in FIG. 12; and

[0041] FIG. 14 is a diagram depicting a flow of another method of synchronization processing (step S80) in FIG. 12.

#### BEST MODE FOR CARRYING OUT THE INVENTION

[0042] Embodiments of the present invention will now be described with reference to the drawings. The embodiments are for assisting in understanding the present invention, and do not limit the technical scope of the present invention.

[0043] FIG. 1 is a configuration example of a game machine to which the present invention is applied.

[0044] The game machine has a game machine main board 1, game machine sub-board 2, IC card reader/writer 3, and a power unit 4. The power unit 4 has a power switch 5 which can be operated from outside the machine.

[0045] The game machine main board 1 has a main CPU 10 and a ROM 12 storing BIOS and the game programs. The game machine main board 1 also has a nonvolatile memory 11 dedicated to the main CPU, where the main CPU 11 reads/writes information during game execution. The main CPU 10 starts up the system by turning the power switch 5 ON, and executing BIOS. When the system is started, an IC card of the player can be inserted into the IC card reader/writer 3.

[0046] When an IC card is inserted into the IC card reader/writer 3, the main CPU 10 reads the card information of the IC card, and authenticates the player. If the player passes authentication, the CPU 10 can start the game program.

[0047] The game machine sub-board 2, on the other hand, has a sub-CPU 20 and a shared nonvolatile memory 21, which is shared with the game machine main board 1. The game machine sub-board 2 functions primarily as an I/O equipment interface, such as for a joystick, and sound amplifier. The game machine sub-board 2 is started up by turning the power switch 5 ON, just like the game machine main board 1.

[0048] In FIG. 1, the sub-CPU 20 of the game machine sub-board 2 has a count function, which is a characteristic of the present invention. When the game machine is started up by turning the power ON, counting is started by this count function, and counting continues until power is shut OFF in the progressing process of the game program, including the final storage of the game result data. And the count value after being counted is stored in the first memory area of the shared nonvolatile memory 21.

[0049] For this count function, it is possible to handle the time change by a clock function in the same way as the counter function, and the main CPU 10 of the game machine main board 1 records a count value acquired based on the count value counted by the count function of the sub-CPU 20. An independent nonvolatile memory may be provided to store this count value, and is used as the second nonvolatile memory area, but it is preferable to store the count value in

a second nonvolatile memory area in the shared nonvolatile memory 21 in the game machine sub-board 2.

[0050] In the following description, a count value counted by the count function of the sub-CPU 20 is called a "life count", and a count value acquired by the main CPU 10 based on this life count is called a "copy count".

[0051] FIG. 2A to 2C shows an example of the relationship between the first memory area and the second memory area.

[0052] In FIG. 2A to 2C, FIG. 2A shows a mode where a life count is stored in a first nonvolatile memory area of the shared nonvolatile memory 21 (step P1), and for the copy count, the main CPU 10 reads the life count (step P2), and copies it to a second nonvolatile memory area of the shared nonvolatile memory 21 (step P3).

[0053] FIG. 2B shows a mode where a life count is stored in a first nonvolatile memory area of the shared nonvolatile memory 21 (step P1), and is simultaneously notified to the main CPU 10 (step P2a). Then the main CPU 10 stores the notified life count in a second nonvolatile memory area of the shared nonvolatile memory 21 (step P3).

[0054] FIG. 2C shows a mode where an independent memory 13 is provided in the game machine main board 1, and a life count is stored in a first nonvolatile memory area of the shared nonvolatile memory 21 (step P1), and is simultaneously notified to the main CPU 10 (step P2a). Then the main CPU 10 stores the notified life count in the independent nonvolatile memory 13 as a second nonvolatile memory area, and acquires a copy count (step P3a).

[0055] FIG. 4 is a diagram depicting a life count and a copy count that the main CPU 10 acquires based on this life count in this configuration, according to the present invention, and this is the case showing that the cause of a restart is a hang up of the main CPU 10.

[0056] In FIG. 4, "A" is a life count and is incremented by "1" each second while the power is ON, as an example. The life count to be incremented is stored in the first nonvolatile memory area of the shared nonvolatile memory 21.

[0057] "B" is a copy count acquired by the main CPU 10 according to one of the methods described in FIG. 2A to 2C. The copy count, which is acquired by the acquisition processing of the main CPU 10 based on the life count by the sub-CPU 20, appears with a delay, as shown in FIG. 4.

[0058] The example in FIG. 4 shows the state when the main CPU 10 hangs up at time T0. Therefore the copy count acquired by the main CPU 10 stops at "101".

[0059] An arcade operator who recognizes this hang up state operates the power switch 5 of the game machine, and shuts the power OFF (t1). At this time, the copy count acquired by the main CPU 10 stops at "101", but in the sub-CPU 20, to which power is continuously supplied after the main CPU 10 hangs up until the power switch 5 is turned OFF, the value of the life count is incremented.

[0060] This generates a difference between the life count value and the copy count value counted by the main CPU 10 and the sub-CPU 20 until the power switch 5 is turned OFF. It can be expected that the difference of the count values of one or more will be generated by the time the arcade operator detects the hang up state of the CPU 10 and shuts



the power switch **5** OFF if the life count is incremented by "1" each second. In other words, in the case of the example in FIG. 4, the life count is "104" at time t1 when the power switch **5** is turned OFF, and the value of the copy count at hang up is "101". Therefore the difference "3" is generated between these count values.

[0061] When the arcade operators turns the power switch **5** ON again, the game machine main board **1** and the sub-board **2** are restarted at times T1 and T2 respectively, as described in FIG. 1. When the game machine main board **1** and the game machine sub-board **2** complete restart respectively, the main CPU **1** compares the counts stored in the first and second nonvolatile memory areas at time T3 according to the game program.

[0062] In the above description, the count values stored in the first and second nonvolatile memory areas are compared by the CPU **1** as the first control unit. In the present invention, however, application is not limited to this. In other words, the CPU **20** may read the count value and the copy count value from the shared nonvolatile memory **21** by the CPU **20** as the second control unit, as shown in FIG. 2A and FIG. 2B, or the copy count value may be read from the independent memory **13** if the copy count value is stored in the independent memory **13**, as shown in FIG. 2C, to compare each value.

[0063] In any case, however, the comparison result must be transferred to the main CPU **10**.

[0064] As another mode, an independent comparison circuit **14** may be disposed, as shown in FIG. 3A to 3C, so as to compare the count value and the copy count value. FIG. 3A, FIG. 3B and FIG. 3C in FIG. 3, correspond to FIG. 2A, FIG. 2B and FIG. 2C respectively. In other words, in the mode shown in FIG. 3A and FIG. 3B, the comparison circuit **14** reads a count value and a copy count value from the shared nonvolatile memory **21**, and compares these values. The result is transferred to the main CPU **1**.

[0065] In the mode shown in FIG. 3C, the comparison circuit **14** reads the count value from the shared nonvolatile memory **21** and the copy count value from the independent memory **13**, and compares these values. The result must also be transferred to the main CPU **1**.

[0066] In this comparison of count values, it is judged that if the difference between the life count which is the first count value, and the copy count which is the second count value, is a predetermined value or more (e.g. two or more), the previous state where the power was shut OFF (previous startup state) is a state where the main CPU **1** was hung up during execution of a program, and no irregular action was committed by players. Therefore the life count and the copy count are reset.

[0067] Application of the present invention was not limited to a method for directly judging the difference when the life count and the copy count are compared, as described above. In other words, the life count value and the copy count value may be set to have a predetermined relationship when the game program is progressing normally, so that whether this predetermined relationship is maintained or not is judged when the life count and the copy count are compared. If the life count value and the copy count value are in the predetermined relationship as a result of comparison, it is judged that irregular action was not committed by players.

[0068] The predetermined relationship refers, for example, to an arithmetic progression, a geometric progression or to a certain relationship to be formed between values before and after conversion using a predetermined function.

[0069] Then incrementing of the life count and acquisition of the copy count by the main CPU **10** are restarted, and also execution of the game program is restarted.

[0070] If the power switch **5** is intentionally turned OFF by an irregular action of a player, on the other hand, incrementing of the counter by the sub-CPU **20** stops immediately. Also the main CPU **10** does not copy the life count. Therefore, in this case, the first count value stored in the first nonvolatile memory area and the second count value stored in the second nonvolatile memory area are the same, or do not maintain the above mentioned predetermined relationship.

[0071] FIG. 5 is a diagram depicting the state of the count values when the power switch **5** is turned OFF by an intentional irregular action of a player, in comparison with FIG. 4.

[0072] In FIG. 5, if the switch **5** is forcibly turned OFF by an intentional irregular action of a player at timing T0', the first and second counter values become the same, "104" for example. Therefore when the arcade operator or a player turns the power switch **5** ON, the game machine main board **1** and the game machine sub-board **2** are restarted at times T1 and T2 respectively. When the game machine main board **1** and the game machine sub-board **2** complete restart respectively, the main CPU **1** compares the first and second count values stored in the first and second nonvolatile memory areas according to the game program at time T3.

[0073] In this comparison of count values, the life count which is the first count value, and the copy count which is the second count value, have no difference, but are identical. Therefore the main CPU **1** judges that the previous state where power is shut OFF is due to an irregular action of a player.

[0074] Various processings are possible after judging that the cause is an irregular action of the player. For example, a record of irregular action is stored in the IC card inserted by a player, so that subsequent participation of this player in a network game is disabled. Or the achievement level in a network game is set to the lowest rank.

[0075] In any case, a disadvantage is given to the player who intentionally shut the system OFF, therefore fairness to other players is insured.

[0076] FIG. 6 is a table summarizing the above modes. The previous startup state can be known by the count state shown in this table.

[0077] In the table in FIG. 6, "I" is a state where the system restarted during normal operation. At most a difference of "1" or less could be generated, depending on the timing of copying the life count. "II" is a state where the game machine main board **1** was hung up, and life count could not be copied. "III" is a state where the life count and the copy count end with the same value. This case shows that the power switch **5** was turned OFF by an intentional irregular action of a player, and the sub-board **2** could not increment the life count.

[0078] In both cases of restarting the system during normal operation and shutting OFF the power switch by an irregular action of a player, the life count and the copy count could either be the same or have a difference of “1” or less. “I” and “III” therefore cannot be distinguished merely by the life count and the copy count.

[0079] Restart in normal operation is performed when the game is not progressing (e.g. during an advertisement), so a flag to indicate whether it is during a game or not is stored in the nonvolatile memory separately from these counters, and this information is referred to at start up, so that “I” and “III” can be distinguished.

[0080] Or whether it is during a game or not can be distinguished by fixing the value to a predetermined value (e.g. 0) without incrementing the life count and the copy count during an advertisement. If it is assumed that the main game board 1 side is normal and the sub-game board 2 side is hung up, the values of the life count and the copy count become the same in theory. However it is rare that the game machine sub-board 2 hangs up, so it is unlikely that this will occur. Therefore it can be concluded that the state in III in FIG. 6 occurred because of a player forcibly shutting power OFF by an irregular action in the previous start up state.

[0081] In the above description, the life count and the copy count are stored in the nonvolatile memory, but it is possible that an undefined value be written when power is turned OFF when writing to the nonvolatile memory. To avoid this problem, as an extension of the present invention, the life count and the copy count are multiplexed.

[0082] FIG. 7 is an example when the data to be written to the nonvolatile memory is triplexed.

[0083] FIG. 8 shows areas in the shared nonvolatile memory 21 corresponding to this triplexing, where three areas for multiplexing are secured for the life count and the copy count respectively. In other words, a same count value is written in different areas in the sequence of count 1, count 2 and count 3, so as to triplex the data.

[0084] In the table in FIG. 7, N is a value which is about to be written now, and N-1 is a value which was written a previous time. X is an arbitrary value, and in a normal operation state X is N or N-1. If the power is shut OFF when writing to the nonvolatile memory, an unexpected value is written.

[0085] A possible method to determine original data from triplexed data being written is a “majority decision”.

[0086] In this method, accessing the memory twice or more times from when power is shut OFF to a complete stop is unlikely to occur. In the case of case 2 in FIG. 7, however, a failure occurs when X is neither N nor N-1. With this problem in view, the present invention decides the original value as follows.

[0087] (1) If 2 or more values, out of counts 1 to 3, are identical, this identical value is judged as the original value of triplexing (see cases 1 and 3).

[0088] (2) If (1) is not established and if the difference between count 3 and count 1 is “1”, then count 3 or count 1 is judged as the original value of triplexing (see case 2).

[0089] In the case of (2), which one of count 3 or count 1 to be used as the original value is not a critical issue. In order

to more closely detect a restart when the game machine main board 1 did not hang up, a smaller value, count 1, is regarded as the original value of the life count, and a greater value, count 3, is regarded as the original value of the copy count in case 2.

[0090] Now the environment where the game machine shown in FIG. 1 is installed will be described. The present invention will be particularly significant when it is applied to the case when a plurality of players play a game simultaneously, or at different times and from different locations.

[0091] Therefore the present invention can be applied in the same way when a plurality of game machines are used to play a game at one arcade, or when a plurality of game machines in a plurality of arcades are used to play a game via a network. In any case, each game machine has a basic configuration shown in FIG. 1.

[0092] FIG. 9 is a configuration example of a game machine system assuming that game machines in a plurality of arcades are used to play a game via a network.

[0093] A plurality of arcades 100A, 100B . . . 100N are connected to a data center 120 via the Internet 110 as a network. The data center 120 has a data base server 121 and stores information on players participating in games by each game title.

[0094] A Web server 122 sends player information in a data base server 121 via a firewall 123 and Internet 110 to each arcade, and collects game execution results from each arcade, and sends it to the data base server 121.

[0095] The data center 120 also has a mail server 124 so as to send player information to a terminal of a registered user (player).

[0096] The arcades 100A, 100B to 100N are connected to the Internet 110 via an arcade router 101 respectively. An arcade management device 102a is connected via an arcade hub 102, and a plurality of game machines 104A to 104N are connected via a hub 103. The configuration of each of the plurality of game machines 104A to 104N is the same as the configuration shown in FIG. 1.

[0097] The arcade management device 102a has a function to manage the power OFF of a game machine and the users. The function to manage power OFF of a game machine and the users may be assigned to the data center 120. Here managing the users refers to controlling how game play is handled, including the rejection of a game continuation for a player who committed an irregular action.

[0098] If an irregular action is taken in one game machine at an arcade and the power of the game machine is shut OFF, the arcade management device 102a acquires and manages information on the player who committed this irregular action. Therefore even if the player who committed the irregular action attempts to continue the game on another game machine in the arcade, execution of the game can be rejected if this player is judged as a player who committed an irregular action before, as a result of player authentication. Or by assigning a predetermined handicap for game execution, fairness to other players can be insured.

[0099] If the arcade management device 102a registers the acquired information on the player who committed an irregular action in the data base server 121 of the data center

**120**, then even if the player who committed the irregular action attempts to execute the game on another game machine of another arcade, the game execution can be rejected, or a predetermined handicap can be assigned to game execution.

[**0100**] Now the actual flow of game processing when the configuration based on the present invention is applied to this game machine system will be described.

[**0101**] FIG. **10** is a diagram depicting the general flow of the game machine system.

[**0102**] When the power of the game machine is turned ON, the game machine main board **1** and the game machine sub-board **2** are started up (step **S1**). The main CPU **10** of the game device main board **1** acquires the previous termination state information (step **S2**). As the termination state information, the life count and the copy count stored in the shared nonvolatile memory **21** are acquired (step **S2**, see FIG. **8**).

[**0103**] Based on the acquired life count and the copy count, the original life count and the copy count are determined using the judgment standard for the life count and the copy count respectively if the counts are multiplexed (example in FIG. **8**).

[**0104**] Using the determined life count and the copy count, it is judged whether the previous termination was caused by a hang up (step **S3**). If the difference of the life count and the copy count is one or more, it is judged as a termination caused by a hang up according to the principle of the present invention, and if there is no difference between the life count and the copy count and the life count and the copy count are identical, then it is judged that the power was shut OFF due to an irregular action by a player.

[**0105**] In FIG. **10**, if it is judged that the previous termination was due to a hang up (step **S3**, **Y**), the recovery data of the play data stored in the nonvolatile memory **11** dedicated to the main CPU **10** is saved (step **S4**). FIG. **11** is an example of play data recorded in the nonvolatile memory **11** dedicated to the main CPU **10**, and the recovery data means saving the data recorded in the recovery data storage area without deleting it.

[**0106**] If it is judged that the previous termination is due to power OFF by an intentional irregular action of a player (step **S3**, **N**), on the other hand, the irregular action of the player+once, for example, is recorded (step **S5**). If the irregular action count does not exceed a threshold (step **S6**, **N**), the restart of the game is enabled, and if the irregular action count exceeds a threshold (step **S6**, **Y**), irregular action countermeasure processing (step **S7**) is executed.

[**0107**] Various handlings are possible for the irregular action countermeasure processing (step **S7**), such as decreasing the ranking of the player data in a network game for a predetermined amount, and rejecting continuation of a game.

[**0108**] Here the player information is not stored in the game machine where an intentional irregular shut off is performed, but in the arcade management device, which is not illustrated, connected to the arcade hub **102** of the arcade or the data server **121** of the data center **120**. This player information can include the irregular action count of the player who committed an intentional irregular action.

[**0109**] In this case, even if the player who committed an irregular action moves to another game machine and attempts to restart the game, a previous irregular action can easily be recognized by checking before starting the game. Therefore in this case as well, game execution of a player who committed an intentional irregular action can be rejected or penalized in an achievement record so as to insure fairness to other players.

[**0110**] If a restart of the game is possible (step **S6**, **N**), the system is initialized (step **S8**). Here an initializing the system refers to synchronizing the game machine main board **1** and the game machine sub-board **2**, and starting the counting of the life count and the copy count.

[**0111**] One possible method is the method shown in FIG. **12**. In other words, as the synchronization processing (step **S80**), a count start command is sent from the main CPU **10** of the game machine main board **1** to the game machine sub-board **2** (step **S80a**). By this, the sub-CPU **20** of the game machine sub-board **2** clears the life count (step **S80b**).

[**0112**] After the life count is cleared, the sub-CPU **20** increments the life count by "1" at each predetermined time (e.g. at each one second) (steps **S81a** and **82a**).

[**0113**] Corresponding to this, the main CPU **10** acquires the copy count at each predetermined time based on the life count (steps **S81b** and **82b**).

[**0114**] FIG. **13** is a flow chart depicting another method of the synchronization processing (step **S80**) in FIG. **12**, and the synchronization is performed on the nonvolatile memory. FIG. **14** is a sequence diagram corresponding to FIG. **13**.

[**0115**] After start up, the game machine main board **1** acquires and confirms the life count and the copy count (step **S2**: FIG. **10**), and when this ends, the game machine main board **1** starts toggling the synchronization data with a predetermined interval (millisecond order) in order to notify the end of acquisition of information of the life count and the copy count to the game machine sub-board **2** (step **800**). For example, **55h**, **AAh**, **55h**, **AAh** . . . is performed.

[**0116**] After start up, the game machine sub-board **2** waits for the synchronization data to change while maintaining the life count in a state before startup (step **S801**). When the synchronization data is changed (step **S801**, **Y**), the game machine sub-board **2** clears the synchronization data to notify the change of the synchronization data to **11** (step **S802**). For clear, any value, other than the values which the game machine main board **1** is toggling, can be used. For example, the synchronization data is cleared to "0".

[**0117**] After this, the life count is cleared to "0" (step **S803**), and regular count up of the life count is started. The game machine main board **1** detects that the synchronization data, which has been toggled, is cleared (step **S804**, **Y**), and starts regular copying of the life count.

[**0118**] This count up (incrementing) processing of the life count by the sub-CPU **20** of the game machine sub-board **2** and the copy count acquisition processing by the main CPU **10** of the game machine main board **1** are continued at least as long as the game machine is operating normally. If the power of the game machine is shut OFF due to a hang up or to irregular action of a player, processing based on the

comparison of the life count and the copy count, which is the characteristic of the present invention described above, is performed.

[0119] Referring back to FIG. 10, when the system initialization (step S8) ends, as mentioned above, normal game machine processing is performed. In other words, an advertisement processing to display an advertisement screen on a display screen, which is not illustrated, of the game machine, is executed (step S9). If a player instructs for the start of the game (step S10, Y), game start processing is performed (step S11), and game execution is enabled (step S12). If game termination is instructed, game termination processing is performed (step S13), and advertisement processing (step S9) is performed again.

[0120] From advertisement processing (step S9) to game termination processing (step S13) are not directly related to the present invention, so further description is omitted.

INDUSTRIAL APPLICABILITY

[0121] By applying the present invention, the cause of a restart can be easily identified, whether the cause is by an intentional irregular action of a player or by a hang up, so fairness to other players participating in a network game can be insured when the restart is due to an intentional irregular action of a player, and diminishing enthusiasm to participate in a game can be prevented.

1. A game machine, comprising:
    - a first control unit controlling execution of a game program;
    - a second control unit continuing incrementing a count value in a progressing process of the game program;
    - a first nonvolatile memory area storing the count value incremented by the second control unit as a life count; and
    - a second nonvolatile memory area storing a count value acquired by the first control unit based on the life count as a copy count, wherein

the life count stored in the first nonvolatile memory area and the copy count stored in the second nonvolatile memory area are compared at restart, and

judgment is made that power is shut OFF due to an irregular action of a player before the restart, if the life count and the copy count are not in a predetermined relationship.
2. The game machine according to claim 1, further comprising a managing unit managing the power shut OFF and a player, wherein
    - when judgment is made that the power is shut OFF due to an irregular action of the player, the management unit controls the player to limit execution of the game.
  3. The game machine according to claim 1, wherein the copy count to be stored in the second nonvolatile memory area is a copy of the life count.
  4. The game machine according to claim 1, wherein the first nonvolatile memory area and the second nonvolatile memory area are in a common nonvolatile memory.
  5. The game machine according to claim 1 or claim 4, wherein the first nonvolatile memory area and the second nonvolatile memory area have a multiplexed area respec-

tively, and the same life count and copy count are written in the multiplexed area respectively.

6. The game machine according to claim 5, wherein if the multiplexed area is a triplexed area and values written in two or more areas of the multiplexed area are the same, then the same value is regarded as the original life count and copy count multiplexed.

7. The game machine according to claim 6, wherein if values written in the two or more areas are not the same, and if a difference of a first value written in a first rank order and a third value written in a third rank order is "1", then the first value or the third value is regarded as the original life count and copy count.

8. A game machine system, comprising a plurality of game machines, and a management device managing power OFF and player information of the plurality of game machines, wherein

each of the plurality of game machines comprises:

- a first control unit controlling execution of a game program;
- a second control unit continuing incrementing a count value in a progressing process of the game program;
- a first nonvolatile memory area storing the count value incremented by the second control unit as a life count; and

a second nonvolatile memory area storing a count value acquired by the first control unit based on the life count as a copy count, and wherein

the life count stored in the first nonvolatile memory area and the copy count stored in the second nonvolatile memory area are compared at restart,

judgment is made that the power is shut OFF due to an irregular action of a player before the restart, if the life count and the copy count are not in a predetermined relationship, and

the management device holds information of the player who committed the irregular action thus judged, and controls the player to limit execution of the game on another game machine.

9. The game machine system according to claim 8, wherein

the management device comprises one data center connected via a network, and

the data center comprises a data base holding information on players who execute a game, and acquires information on a player who committed an irregular action on the plurality of game machines, which is transmitted via the management device, and registers the information in the data base.

10. A method for detecting an irregular action of a player, comprising the steps of:

- continuing incrementing a count value by a second control unit in a progressing process of a game program which is controlled for execution by a first control unit;
- storing a count value incremented by the second control unit in a first nonvolatile memory as a life count;
- acquiring a count value by the first control unit based on the life count;

storing the count value acquired by the first control unit in a second nonvolatile memory as a copy count;

comparing the life count stored in the first nonvolatile memory area and the copy count stored in the second nonvolatile memory area at restart; and

judging that the power is shut OFF due to an irregular action of a player before the restart if the life count and copy count are not in a predetermined relationship.

11. A storage medium readable in a game machine, storing a program for detecting an irregular action of a player, when the program is executed, the game machine performing the steps of:

continuing incrementing a count value by a second control unit in a progressing process of a game program which is controlled for execution by a first control unit;

storing a count value incremented by the second control unit in a first nonvolatile memory as a life count;

acquiring a count value by the first control unit based on the life count;

storing the count value acquired by the first control unit in a second nonvolatile memory as a copy count;

comparing the life count stored in the first nonvolatile memory area and the copy count stored in the second nonvolatile memory area at restart; and

judging that the power is shut OFF due to an irregular action of a player before the restart if the life count and copy count are not in a predetermined relationship.

\* \* \* \* \*