(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 4 113 305 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent: **05.06.2024 Bulletin 2024/23**

(21) Application number: **21181992.5**

(22) Date of filing: **28.06.2021**

(51) International Patent Classification (IPC):
G06F 11/14 $^{(2006.01)}$ ***G06F 11/36*** $^{(2006.01)}$

(52) Cooperative Patent Classification (CPC):
G06F 11/3636

(54) REPRODUCING A STATE OF A SYSTEM AND A NETWORK UPON OCCURRENCE OF AN ERROR

WIEDERGABE EINES ZUSTANDES EINES SYSTEMS UND EINES NETZWERKS BEI AUFTRETEN EINES FEHLERS

REPRODUCTION DE L'ÉTAT D'UN SYSTÈME ET D'UN RÉSEAU LORS DE L'APPARITION D'UNE ERREUR

(84) Designated Contracting States:
AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR

(43) Date of publication of application:
04.01.2023 Bulletin 2023/01

(73) Proprietor: **Elektrobit Automotive GmbH**
91058 Erlangen (DE)

(72) Inventor: **Asemann, Peter**
60488 Frankfurt am Main (DE)

(74) Representative: **Continental Corporation**
c/o Continental Automotive Technologies GmbH
Intellectual Property
Guerickestraße 7
60488 Frankfurt a. Main (DE)

(56) References cited:
US-A1- 2011 078 666 **US-A1- 2015 378 870**
US-B1- 8 117 600

Note: Within nine months of the publication of the mention of the grant of the European patent in the European Patent Bulletin, any person may give notice to the European Patent Office of opposition to that patent, in accordance with the Implementing Regulations. Notice of opposition shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

Processed by Luminess, 75001 PARIS (FR)

Description

[0001] The present invention is related to a method, a computer program code, and an apparatus for reproducing a state of a system and a network upon occurrence of an error.

[0002] In the field of embedded computing a problem exists in that occurrence of an error is detected, e.g. from logs of the network traffic, but the error cannot be reproduced and analyzed once it has occurred. As there could be many reasons why operation of a system failed, e.g. operation of a processor or a system-on-chip, it is common that a group of developers tries to reproduce the issue for weeks or months before it is spotted again. This is a very cumbersome process and a huge issue for errors that appear only sporadically.

[0003] US 2015/0378870 A1 discloses a solution for time travel debugging in a managed runtime system. The managed runtime system includes an execution component that executes a managed program component. Moreover, the managed runtime system includes a time travel debugger component. The time travel debugger component is configured to record a sequence of live-object snapshots of program states during execution of the managed program component. A live-object snapshot includes live objects from a heap in memory at a given time during the execution. Moreover, the time travel debugger component is configured to replay at least a portion of the execution of the managed program component based upon the live-object snapshots.

[0004] US 2011/0078666 A1 discloses a method for reproducing electronic program execution. The method comprises running a program and collecting output data while the program is running. The method further comprises performing an output deterministic execution, searching a predetermined space of potential executions of the program, and calculating inferences from the collected output data to find operational errors in the program.

[0005] US 8,117,600 B1 discloses a method for identifying in-line synchronization instructions in binary program code. Executable segments of the binary program code are scanned to identify potential in-line synchronization instructions. For each potential in-line synchronization instruction, it is determined whether neighboring potential instructions are valid instructions. For each potential in-line synchronization instruction, it is determined that the potential in-line synchronization instruction is a valid in-line synchronization instruction if the neighboring potential instructions are valid instructions.

[0006] It is an object of the present invention to provide an improved solution for reproducing a state of a system and a network upon occurrence of an error.

[0007] In accordance with the invention, this object is achieved by a method according to claim 1, by a computer program code according to claim 10, which implements this method, and by an apparatus according to claim 11. The dependent claims include advantageous further developments and improvements of the present principles as described below.

[0008] According to a first aspect, a method for reproducing a state of a system and a network upon occurrence of an error comprises the steps of:

- continuously recording a state of the system for a time window by a tracing debugger;
- generating a network log for the network by a network tracer;
- detecting a network condition indicative of an error or an internal condition indicative of an error;
- storing a recorded state of the system before the error together with a log of network system actions until appearance of the error;
- restoring the recorded state before the error on the system; and
- replaying logged network traffic to reproduce the error.

[0009] Accordingly, a computer program code comprises instructions, which, when executed by at least one processor, cause the at least one processor to perform the following steps for reproducing a state of a system and a network upon occurrence of an error:

- continuously recording a state of the system for a time window by a tracing debugger;
- generating a network log for the network by a network tracer;
- detecting a network condition indicative of an error or an internal condition indicative of an error;
- storing a recorded state of the system before the error together with a log of network system actions until appearance of the error;
- restoring the recorded state before the error on the system; and
- replaying logged network traffic to reproduce the error.

[0010] The term computer has to be understood broadly. In particular, it also includes workstations, embedded devices and other processor-based data processing devices.

[0011] The computer program code can, for example, be made available for electronic retrieval or stored on a computer-readable storage medium.

[0012] According to another aspect, an apparatus for reproducing a state of a system and a network upon occurrence of an error comprises:

- a system analyzing means configured to continuously record a state of the system for a time window and to detect an internal condition indicative of an error, wherein the analyzing means is a tracing debugger;
- a network analyzing means configured to generate a network log for the network and to detect a network condition indicative of an error, wherein the network

analyzing means is a network tracer. and

- a storage device configured to store a recorded state of the system before the error together with a log of network system actions until appearance of the error;

wherein the apparatus is configured to restore the recorded state before the error on the system and to replay logged network traffic to reproduce the error.

[0013] The invention combines continuously recording an image of the inner state of a system, e.g. a window of some seconds or optionally the whole development of the state from the time the system is started, with recording network traffic. In case an error is detected, a relevant part of the recorded information is frozen and persisted. This allows reproducing and analyzing the error conditions or faults inside the system, which caused the occurrence of the error. To this end, a recorded state is restored on the system and logged network traffic of remote communication partners is replayed. By restoring a recorded state before the error on the system and replaying the logged network traffic, it can be checked if the resulting inner working of the system matches the recorded system trace and if the system reacts in the same way as in the network log. If this is the case, the scenario has been restored successfully and the error can be reproduced repeatedly. An advantage of the described solution is that error conditions becoming visible in the network traffic or being visible for the system can be analyzed and debugged and possibly reproduced right away, without a need to wait for the issue to reappear.

[0014] The network log for the network is generated by a network tracer. Network tracers are established tools for analyzing network traffic and are readily available for a wide range of network types. Preferably, the network condition indicative of an error is detected by the network tracer. In this way, there is no need to provide an additional module or software component for evaluating the network conditions.

[0015] The state of the system is recorded by a tracing debugger. Such tracing debuggers are able to record and store considerable parts of the internal state of a system, e.g. of an electronic control unit.

[0016] In an advantageous embodiment, the network condition indicative of an error includes at least one of a duplicate packet, a broken packet, and a packet not matching an assumed protocol state. For example, a broken packet may be determined from at least one of a wrong cyclic redundancy code, a wrong length, and a wrong message authentication code. Packets not matching an assumed protocol state can easily be detected if the network tracer can keep track of the protocol state or calculate the protocol state using a suitable model. An example for an assumed protocol state is connection open/closed in case of the transmission control protocol (TCP).

[0017] In an advantageous embodiment, an internal condition indicative of an error includes at least one of an exception, an illegal peripheral input value, and an error handling code being executed. In this way, not only errors visible in the network trigger the described mechanism, but also errors that are visible and relevant to the system in some way, such as errors in the directly connected peripherals, processor-internal errors and errors handled by the software.

[0018] In an advantageous embodiment, the state of the system is recorded in the form of snapshots and/or incremental logs. Both approaches enable to reconstruct the system state at a desired point in time in a reliable way.

[0019] In an advantageous embodiment, the recorded state of the system includes memory accesses, task changes, and code executions. This information will in general be sufficient to describe the inner working of the system.

[0020] In an advantageous embodiment, the network tracer is configured to signal the tracing debugger to store the state of the system at a defined point in time before the detected network condition indicative of an error and a state trace of the system until the point in time the network condition indicative of an error was detected. Alternatively, the system itself is configured to signal the tracing debugger to store the state of the system at a defined point in time before the internal condition indicative of an error was detected and a state trace of the system until the point in time the internal condition indicative of an error was detected. Likewise, the tracing debugger may be configured to detect the internal condition indicative of an error and to store the state of the system at a defined point in time before the internal condition indicative of an error was detected and a state trace of the system until the point in time the internal condition indicative of an error was detected. In this way, it is ensured that a usable system state and a relevant trace of the system state are available for a later analysis.

[0021] In an advantageous embodiment, information on network activity and inner working of the system is provided for analysis. The inner working of the system, i.e. the stored system state and the relevant trace of the system state, as well as the network activity, i.e. the network state and the trace of the network state, are preferably displayed in parallel for analysis. In this way, it is possible to pinpoint the specific condition that led to an error.

[0022] In an advantageous embodiment, the recorded state is restored on the system with a modified software code. In this way, it is possible to test software fixes with all circumstances except for the modified code staying exactly the same. This dramatically reduces the time required for testing software fixes.

[0023] Further features of the present invention will become apparent from the following description and the appended claims in conjunction with the figures.

Figures

[0024]

Fig. 1 schematically illustrates a method for reproducing a state of a system and a network upon occurrence of an error

Fig. 2 schematically illustrates a first embodiment of an apparatus for reproducing a state of a system and a network upon occurrence of an error

Fig. 3 schematically illustrates a second embodiment of an apparatus for reproducing a state of a system and a network upon occurrence of an error

Fig. 4 schematically illustrates a system diagram of a solution according to the invention for reproducing a state of a system and a network upon occurrence of an error

Fig. 5 schematically illustrates an analysis of a fault

Fig. 6 schematically illustrates a live reproduction of a fault in the original system

Fig. 7 schematically illustrates buffers used for recording the state of the system; and

Fig. 8 schematically illustrates a process used for recording the state of the system.

Detailed description

[0025] The present description illustrates the principles of the present disclosure. It will thus be appreciated that those skilled in the art will be able to devise various arrangements that, although not explicitly described or shown herein, embody the principles of the disclosure.

[0026] All examples and conditional language recited herein are intended for educational purposes to aid the reader in understanding the principles of the disclosure and the concepts contributed by the inventor to furthering the art, and are to be construed as being without limitation to such specifically recited examples and conditions.

[0027] Moreover, all statements herein reciting principles, aspects, and embodiments of the disclosure, as well as specific examples thereof, are intended to encompass both structural and functional equivalents thereof.

[0028] Thus, for example, it will be appreciated by those skilled in the art that the diagrams presented herein represent conceptual views of illustrative circuitry embodying the principles of the disclosure.

[0029] The functions of the various elements shown in the figures may be provided through the use of dedicated hardware as well as hardware capable of executing software in association with appropriate software. When provided by a processor, the functions may be provided by a single dedicated processor, by a single shared processor, or by a plurality of individual processors, some of which may be shared. Moreover, explicit use of the term "processor" or "controller" should not be construed to refer exclusively to hardware capable of executing software, and may implicitly include, without limitation, digital signal processor (DSP) hardware, systems on a chip, microcontrollers, read only memory (ROM) for storing software, random access memory (RAM), and nonvolatile storage.

[0030] Other hardware, conventional and/or custom, may also be included. Similarly, any switches shown in the figures are conceptual only. Their function may be carried out through the operation of program logic, through dedicated logic, through the interaction of program control and dedicated logic, or even manually, the particular technique being selectable by the implementer as more specifically understood from the context.

[0031] In the claims hereof, any element expressed as a means for performing a specified function is intended to encompass any way of performing that function including, for example, a combination of circuit elements that performs that function or software in any form, including, therefore, firmware, microcode or the like, combined with appropriate circuitry for executing that software to perform the function. The disclosure as defined by such claims resides in the fact that the functionalities provided by the various recited means are combined and brought together in the manner which the claims call for. It is thus regarded that any means that can provide those functionalities are equivalent to those shown herein.

[0032] Fig. 1 schematically illustrates a method according to the invention for reproducing a state of a system and a network upon occurrence of an error. A state of the system is continuously recorded S1 for a time window by a tracing debugger. Advantageously, the state is recorded in the form of snapshots and/or incremental logs and includes memory accesses, task changes, and code executions. Furthermore, a network log for the network is generated S2 by a network tracer. When a network condition indicative of an error is detected S3, preferably by the network tracer, or an internal condition indicative of an error is detected S3, preferably by the tracing debugger, at least a part of the recorded state of the system is stored S4 together with a fitting portion of the network log for the network. The network condition indicative of an error may include, for example, at least one of a duplicate packet, a broken packet, and a packet not matching an assumed protocol state. A broken packet is advantageously determined from at least one of a wrong cyclic redundancy code, a wrong length, and a wrong message authentication code. Advantageously, the network tracer is configured to signal the tracing debugger to store the state of the system at a defined point in time before the detected network condition indicative of an error and a state trace of the system until the point in time the network condition indicative of an error was detected

S3.

[0033] Alternatively, an internal condition indicative of an error, such as a processor exception, an invalid input, or an internal error handling code being executed, can trigger the tracing debugger to store the state of the system at a defined point in time before the detected internal condition indicative of an error and a state trace of the system until the point in time the internal condition indicative of an error was detected S3. Optionally, information on network activity and inner working of the system may be provided S5 for analysis. Furthermore, a recorded state is restored S6 on the system and logged network traffic of remote communication partners is replayed S7. If desired, the recorded state may be restored S6 on the system with a modified software code, e.g. for testing software fixes.

[0034] Fig. 2 schematically illustrates a block diagram of a first embodiment of an apparatus 20 according to the invention for reproducing a state of a system S and a network N upon occurrence of an error. The apparatus 20 has an input 21 for receiving data from the system S and the network N. The apparatus 20 further has a system analyzing means 22, i.e. a tracing debugger, configured to continuously record a state of the system for a time window. Preferably, the system analyzing means 22 is configured to detect internal conditions indicative of an error, such as processor exceptions, illegal input values, or the execution of an error handling code, and store the state of the system S at a defined point in time before the detected internal condition indicative of an error and a state trace of the system S until the point in time the internal condition indicative of an error was detected.

[0035] Advantageously, the system analyzing means 22 is configured to record the state in the form of snapshots and/or incremental logs and includes memory accesses, task changes, and code executions. A network analyzing means 23, i.e. a network tracer, is configured to generate a network log for the network and to detect a network condition indicative of an error. The network condition indicative of an error may include, for example, at least one of a duplicate packet, a broken packet, and a packet not matching an assumed protocol state. A broken packet is advantageously determined from at least one of a wrong cyclic redundancy code, a wrong length, and a wrong message authentication code. A storage device 24 is provided for storing at least a part of the recorded state of the system together with a fitting portion of the network log for the network. Advantageously, the network analyzing means 23 is configured to signal the system analyzing means 22 to store the state of the system S at a defined point in time before the detected network condition indicative of an error and a state trace of the system S until the point in time the network condition indicative of an error was detected. Data stored on the storage device 24 may be made available for analysis via an output 26. Advantageously, the apparatus 20 is configured to restore a recorded state on the system S and to replay logged network traffic of remote communication partners. If desired, the recorded state may be restored on the system S with a modified software code, e.g. for testing software fixes.

[0036] The system analyzing means 22 and the network analyzing means 23 may be controlled by a control module 25. A user interface 27 may be provided for enabling a user to modify settings of the system analyzing means 22, the network analyzing means 23, and the control module 25. The system analyzing means 22, the network analyzing means 23, and the control module 25 can be embodied as dedicated hardware units. Of course, they may likewise be fully or partially combined into a single unit or implemented as software running on a processor, e.g. a CPU or a GPU.

[0037] A block diagram of a second embodiment of an apparatus 30 according to the invention for reproducing a state of a system and a network upon occurrence of an error is illustrated in Fig. 3. The apparatus 30 comprises a processing device 32 and a memory device 31. For example, the apparatus 30 may be a computer, an electronic control unit or an embedded system. The memory device 31 has stored instructions that, when executed by the processing device 32, cause the apparatus 30 to perform steps according to one of the described methods. The instructions stored in the memory device 31 thus tangibly embody a program of instructions executable by the processing device 32 to perform program steps as described herein according to the present principles. The apparatus 30 has an input 33 for receiving data. Data generated by the processing device 32 are made available via an output 34. In addition, such data may be stored in the memory device 31. The input 33 and the output 34 may be combined into a single bidirectional interface.

[0038] The processing device 32 as used herein may include one or more processing units, such as microprocessors, digital signal processors, or a combination thereof.

[0039] The storage device 24 and the memory device 31 may include volatile and/or non-volatile memory regions and storage devices such as hard disk drives, optical drives, and/or solid-state memories.

[0040] In the following, further details of the invention shall be given with reference to Fig. 4 to Fig. 8.

[0041] Fig. 4 schematically illustrates a system diagram of a solution according to the invention for reproducing a state of a system S and a network N upon occurrence of an error. In this example, the system S is a system-on-chip comprising a random access memory RAM, a number of registers REG, and two processing cores C0, C1. The system S is connected to a network N via a bus B. A tracing debugger TD performs a continuous transparent tracing of the system state. For this purpose, the tracing debugger TD comprises a buffer BUF, e.g. a dual ring buffer. A network tracer NT performs a continuous transparent tracing of network communication via a test access point (TAP). The network tracer NT makes use of a set of anomaly detection rules ADR for detecting a network condition indicative of an error. The tracing debugger TD may monitor if internal error conditions apply in the system-on-chip peripherals, if an internal error handling code is executed, which can be detected using watch points or breakpoints, or if an exception handler is called by a processing core. In case such a condition is detected, the trace of the system state is frozen and persisted together with a fitting porting of the network log of the detected error. For this purpose, the network tracer NT provides a signal SIG to the tracing debugger TD, which indicates a point in time at which to persist the system state. The result is a joint image JI of the state of the system S before the fault and a log of network system actions until appearance of the fault.

[0042] Fig. 5 schematically illustrates an analysis of a fault. The joint image JI of the state of the system S before the fault and a log of network system actions until appearance of the fault is made available for analysis, e.g. to a group of developers in charge of pinpointing the error condition. For this purpose, the network activity and the inner working of the system S over time t, including all memory accesses, task changes, and code executions, are displayed in parallel by an overalls system viewer SV.

[0043] Fig. 6 schematically illustrates a live reproduction of a fault in the original system S. The joint image JI of the state of the system S before the fault and a log of network system actions until appearance of the fault is used for restoring a recorded state on the system S and for replaying logged network traffic of remote communication partners. In this way, it can be checked if the resulting inner working of the system matches the recorded system trace and if the system reacts in the same way as in the network log. If this is the case, the scenario has been restored successfully and the error can be reproduced repeatedly. In order to test software fixes, the recorded state can be restored on the system S with a modified software code. In this way, all circumstances except for the modified code stay exactly the same. This dramatically reduces the time required for testing software fixes.

[0044] Fig. 7 schematically illustrates buffers RBS, RBT used for recording the state of the system. For recording of the complete system state a set of two ring buffers RBS, RBT is preferably used. The first ring buffer RBS is provided for complete system images 11-IN, whereas the second ring buffer RBT is provided for recording trace data TD1-TDN. Both ring buffers RBS, RBT have at least two pages. The number of pages is identical for both ring buffers RBS, RBT. The page sizes are chosen such that each page of the first ring buffer RBS can hold one complete image 11-IN of the state of the system and each page of the second ring buffer RBT can hold some seconds of the trace TD1-TDN of the changes to the state of the system. In addition, a continuous log LOG of network events and peripheral input is generated, i.e. a network I/O trace. All data is preferably timestamped to enable the reconstruction of the system state by reproducing each change in the system.

[0045] Fig. 8 schematically illustrates an exemplary process used for recording the state of the system. In this example, continuously recording the state of the system is achieved as follows. First, the initial image of the state of the system is stored in the active storage page of the first ring buffer. Then the system is started and all changes to the system state are recorded, e.g. by a tracing debugger or similar means, and stored in the active page of the second ring buffer. If the active storage page of the second ring buffer is full, another page becomes the active page, and the recording of the state of the system continues in the new active page. In addition, a complete image In+1 of the state of the system at the moment when recording system changes switches to the new active page is calculated from the image In in the inactive page and all state changes stored as a trace tdn in the inactive page. This image In+1 is stored in the new active page of the first ring buffer. This process is continuously repeated. If no empty pages are available any more for the images and the trace data, the oldest pages are deleted and the recording process continues. This is done to reduce the amount of required storage, if storage is limited. By continuously recording complete images of the state of the system as well as trace data, the ring buffers always contains one or more recent images of the state of the system and a trace of the changes which took place after the point in time represented by the respective images of the state of the system. If an error is detected, all pages of the ring buffer are persisted in persistent memory, e.g. a hard disk, for later reproduction of the error condition.

[0046] If an error is detected at a given point in time Terr, the state of the system can be restored for any point in time Tx which is more recent than the earliest available image, by either taking the most recent image before this point in in time Tx and applying the changes recorded in the trace data, or by replaying the network I/O input recorded in the continuous log of network events and peripheral input from the time the used image was stored to time Tx. If both methods result in the same image, it can safely be assumed that the reconstructed state of the system at time Tx is accurate. The reconstructed state of the system at time Tx can, therefore, be taken as a starting point to reproduce the error at time Terr. By starting from this reconstructed state of the system and replaying the rest of the continuous log of network events and peripheral input reproduction of the error will most likely succeed. In case the error does not appear again, the saved trace buffer data may help to analyze why the error cannot be reproduced. This may help to find or catch spurious hardware errors or issues that are extremely critical with regard to timing.

Reference numerals

[0047]

20	Apparatus
21	Input
22	System analyzing means
23	Network analyzing means
24	Storage device
25	Control module
26	Output
27	User interface
30	Apparatus
31	Memory device
32	Processing device
33	Input
34	Output

ADR	Anomaly detection rule
B	Bus
BUF	Buffer
C0, C1	Core
I_1,..., I_n,	System state image
I_{n+1},..., IN JI	Joint image
LOG	Log of network and peripheral events
N	Network
NT	Network tracer
R	Rest of bus
RAM	Random access memory
RBS	Ring buffer for system state images
RBT	Ring buffer for trace data
REG	Register
S	System
SIG	Signal
SV	System viewer
TD_1 ... TD_N	Trace data
TD	Tracing debugger

S1	Record state of system
S2	Generate network log
S3	Detect network condition or internal condition indicative of error
S4	Store part of recorded state and portion of network log
S5	Provide stored information for analysis
S6	Restore recorded state
S7	Replay logged network traffic

Claims

1. A method for reproducing a state of a system (S) and a network (N) upon occurrence of an error, the method comprising:

- continuously recording (S1) a state of the system (S) for a time window, wherein the state of the system (S) is recorded (S1) by a tracing debugger (TD);
- generating (S2) a network log for the network (N), wherein the network log for the network (N) is generated (S2) by a network tracer (NT);
- detecting (S3) a network condition indicative of an error or an internal condition indicative of an error;
- storing (S4) a recorded state of the system (S) before the error together with a log of network system actions until appearance of the error;
- restoring (S6) the recorded state before the error on the system (S); and
- replaying (S7) logged network traffic to reproduce the error. U

2. The method according to claim 1, wherein the network condition indicative of an error is detected (S3) by the network tracer (NT).

3. The method according to claim 1 or 2, wherein the network condition indicative of an error includes at least one of a duplicate packet, a broken packet, and a packet not matching an assumed protocol state.

4. The method according to one of claims 1 to 3, wherein an internal condition indicative of an error includes at least one of an exception, an illegal peripheral input value, and an error handling code being executed.

5. The method according to one of the preceding claims, wherein the state of the system (S) is recorded (S1) in the form of snapshots and/or incremental logs.

6. The method according to one of the preceding claims, wherein the recorded state of the system (S) includes memory accesses, task changes, and code executions.

7. The method according to one of the preceding claims, wherein the network tracer (NT) is configured to signal the tracing debugger (TD) to store the state of the system (S) at a defined point in time before the network condition indicative of an error was detected (S3) and a state trace of the system (S) until the point in time the network condition indicative of an error was detected (S3), or wherein the system (S) is configured to signal the tracing debugger (TD) to store the state of the system at a defined point in time before the internal condition indicative of an error was detected (S3) and a state trace of the system until the point in time the internal condition indicative of an error was detected (S3), or wherein the tracing debugger (TD) is configured to detect (S3) the internal condition indicative of an error and to store the state of the system at a defined point in time before the internal condition indicative of an error was detected (S3) and a state trace of the system until the point in time the internal condition indicative of an error was detected (S3).

8. The method according to one of the preceding claims, further comprising providing (S5) information on network activity and inner working of the system (S) for analysis.

9. The method according to one of the preceding claims, wherein the recorded state is restored on the system (S) with a modified software code.

10. A computer program code comprising instructions, which, when executed by at least one processor, cause the at least one processor to perform a method according to any of claims 1 to 9 for reproducing a state of a system (S) and a network (N) upon occurrence of an error.

11. An apparatus (20) for reproducing a state of a system (S) and a network (N) upon occurrence of an error, the apparatus (20) comprising:

- a system analyzing means (22) configured to continuously record (S1) a state of the system (S) for a time window, wherein the state of the system (S) is recorded (S1) by a tracing debugger (TD);
- a network analyzing means (23) configured to generate (S2) a network log for the network (N) and to detect (S3) a network condition indicative of an error; wherein the network log for the network (N) is generated (S2) by a network tracer (NT); U and
- a storage device (24) configured to store (S4) a recorded state of the system (S) before the error together with a log of network (N) system actions until appearance of the error;

wherein the apparatus is further configured to restore (S6) the recorded state before the error on the system (S) and to replay (S7) logged network traffic to reproduce the error. U

Patentansprüche

1. Verfahren zum Wiedergeben eines Zustands eines Systems (S) und eines Netzwerks (N) nach einem Auftreten eines Fehlers, wobei das Verfahren umfasst:

- kontinuierliches Aufzeichnen (S1) eines Zustands des Systems (S) während eines Zeitfensters, wobei der Zustand des Systems (S) durch einen Verfolgungs-Debugger (Tracing Debugger, TD) aufgezeichnet wird (S1);
- Erzeugen (S2) eines Netzwerkprotokolls für das Netzwerk (N), wobei das Netzwerkprotokoll für das Netzwerk (N) durch einen Netzwerkverfolger (Network Tracer, NT) erzeugt wird (S2);
- Erkennen (S3) einer Netzwerkbedingung, die einen Fehler oder eine interne Bedingung anzeigt, die einen Fehler anzeigt;
- Speichern (S4) eines aufgezeichneten Zustands des Systems (S) vor dem Fehler zusammen mit einem Protokoll der Netzwerksystemaktionen bis zu dem Erscheinen des Fehlers;
- Wiederherstellen (S6) des aufgezeichneten Zustands vor dem Fehler in dem System (S); und
- Wiederholen (S7) des protokollierten Netzwerkdatenverkehrs, um den Fehler wiederzugeben.

2. Verfahren nach Anspruch 1, wobei die Netzwerkbedingung anzeigt, dass durch den Netzwerkverfolger (NT) ein Fehler entdeckt wurde (S3).

3. Verfahren nach Anspruch 1 oder 2, wobei die Netzwerkbedingung, die einen Fehler anzeigt, mindestens eine eines duplizierten Datenpakets, eines beschädigten Datenpakets und eines Datenpakets anzeigt, das nicht mit einem vorgegebenen Protokollzustand übereinstimmt.

4. Verfahren nach einem der Ansprüche 1 bis 3, wobei eine interne Bedingung anzeigt, dass ein Fehler mindestens einen von einer Ausnahme, einem illegalen peripheren Eingabewert und einem Fehlerbehandlungscode anzeigt, der ausgeführt wird.

5. Verfahren nach einem der vorhergehenden Ansprüche, wobei der Zustand des Systems (S) in der Form von Schnappschüssen und/oder inkrementellen Protokollen aufgezeichnet wird (S1),

6. Verfahren nach einem der vorhergehenden Ansprüche, wobei der aufgezeichnete Zustand des Systems (S) Speicherzugriffe, Aufgabenänderungen und Codeausführungen beinhaltet.

7. Verfahren nach einem der vorhergehenden Ansprüche, wobei der Netzwerkverfolger (NT) konfiguriert ist, um dem Verfolgungs-Debugger (TD) zu signalisieren, dass er den Zustand des Systems (S) zu einem definierten Zeitpunkt, bevor die Netzwerkbedingung, die einen Fehler anzeigt, erkannt wurde (S3) und eine Zustandsverfolgung des Systems (S) bis zu dem Zeitpunkt speichert, an dem die Netzwerkbedingung, die einen Fehler anzeigt, erkannt wurde (S3), oder wobei das System (S) konfiguriert ist, um dem Verfolgungs-Debugger (TD) zu signalisieren, dass er den Zustand des Systems zu einem definierten Zeitpunkt, bevor die interne Bedingung, die einen Fehler anzeigt, erkannt wurde (S3), und eine Zustandsverfolgung des Systems bis zu dem Zeitpunkt speichert, an dem die interne Bedingung, die einen Fehler anzeigt, erkannt wurde (S3), oder wobei der

Verfolgungs-Debugger (TD) konfiguriert ist zum Erkennen (S3) der internen Bedingung, die einen Fehler anzeigt, und zum Speichern des Zustands des Systems zu einem definierten Zeitpunkt, bevor die interne Bedingung, die den Fehler anzeigt, erkannt wurde (S3), und einer Zustandsverfolgung des Systems bis zu dem Zeitpunkt, an dem die interne Bedingung, die den Fehler anzeigt, erkannt wurde (S3).

8. Verfahren nach einem der vorhergehenden Ansprüche, das ferner ein Bereitstellen (S5) von Informationen über die Netzwerkaktivität und eine innere Funktion des Systems (S) für eine Analyse umfasst.

9. Verfahren nach einem der vorhergehenden Ansprüche, wobei der aufgezeichnete Zustand in dem System (S) mit einem modifizierten Softwarecode wiederhergestellt wird.

10. Computerprogrammcode, der Anweisungen umfasst, die, wenn sie von mindestens einem Prozessor ausgeführt werden, den mindestens einen Prozessor veranlassen, ein Verfahren nach einem der Ansprüche 1 bis 9 zum Wiedergeben eines Zustands eines Systems (S) und eines Netzwerks (N) nach einem Auftreten eines Fehlers durchzuführen.

11. Einrichtung (20) zum Wiedergeben eines Zustands eines Systems (S) und eines Netzwerks (N) nach einem Auftreten eines Fehlers, wobei die Einrichtung (20) umfasst:

- ein Element zum Analysieren eines Systems (22), das konfiguriert ist zum kontinuierlichen Aufzeichnen (S1) eines Zustands des Systems (S) während eines Zeitfensters, wobei der Zustand des Systems (S) durch einen Verfolgungs-Debugger (Tracing Debugger, TD) aufgezeichnet wird (S1);
- ein Element zum Analysieren eines Netzwerks (23), das konfiguriert ist zum Erzeugen (S2) eines Netzwerkprotokolls für das Netzwerk (N) und zum Erkennen (S3) einer Netzwerkbedingung, die einen Fehler anzeigt; wobei das Netzwerkprotokoll für das Netzwerk (N) durch einen Netzwerkverfolger (Network Tracer, NT) erzeugt wird (S2);
- eine Speichervorrichtung (24), die konfiguriert ist zum Speichern (S4) eines aufgezeichneten Zustands des Systems (S) vor dem Fehler zusammen mit einem Protokoll der Systemaktionen des Netzwerks (N) bis zu dem Auftreten des Fehlers;

wobei die Einrichtung ferner konfiguriert ist zum Wiederherstellen (S6) des aufgezeichneten Zustands vor dem Fehler in dem System (S); und zum Wiederholen (S7) des protokollierten Netzwerkdatenverkehrs, um den Fehler wiederzugeben.

Revendications

1. Procédé pour reproduire l'état d'un système (S) et d'un réseau (N) lors de l'apparition d'une erreur, le procédé comprenant les étapes suivantes :

- enregistrer de manière continue (S1) l'état du système (S) pendant une fenêtre temporelle, l'état du système (S) étant enregistré (S1) par un débogueur avec traçage (TD) ;
- générer (S2) un journal de réseau pour le réseau (N), le journal de réseau pour le réseau (N) étant généré (S2) par un traceur de réseau (NT) ;
- détecter (S3) une condition de réseau indicative d'une erreur ou d'une condition interne indicative d'une erreur ;
- stocker (S4) un état enregistré du système (S) avant l'erreur ainsi qu'un journal des actions système de réseau jusqu'à l'apparition de l'erreur ;
- restaurer (S6) l'état enregistré avant l'erreur sur le système (S) ; et
- rejouer (S7) le trafic réseau enregistré pour reproduire l'erreur.

2. Procédé selon la revendication 1, dans lequel l'état du réseau indicatif d'une erreur est détecté (S3) par le traceur de réseau (NT).

3. Procédé selon la revendication 1 ou la revendication 2, dans lequel l'état du réseau indicatif d'une erreur comprend au moins l'un parmi un paquet dupliqué, un paquet cassé, et un paquet ne correspondant pas à un état de protocole supposé.

4. Procédé selon l'une des revendications 1 à 3, dans lequel un état interne indicatif d'une erreur comprend au moins un élément parmi une exception, une valeur d'entrée périphérique illégale et un code de gestion d'erreur en cours d'exécution.

5. Procédé selon l'une des revendications précédentes, dans lequel l'état du système (S) est enregistré (S1) sous forme d'instantanés et/ou de journaux incrémentiels.

6. Procédé selon l'une des revendications précédentes, dans lequel l'état enregistré du système (S) comprend des accès à la mémoire, des changements de tâches et des exécutions de codes.

7. Procédé selon l'une des revendications précédentes, dans lequel le traceur de réseau (NT) est configuré pour signaler au débogueur avec traçage (TD) de stocker l'état du système (S) à un moment défini avant que la condition de réseau indicative d'une erreur ait été détectée (S3) et une trace d'état du système (S) jusqu'au moment où la condition de réseau indicative qu'une erreur a été détectée (S3), ou dans lequel le système (S) est configuré pour signaler au débogueur avec traçage (TD) de stocker l'état du système à un moment défini avant que la condition interne indicative d'une erreur ait été détectée (S3) et une trace d'état du système jusqu'au moment où la condition interne indicative d'une erreur a été détectée (S3), ou dans lequel le débogueur avec traçage (TD) est configuré pour détecter (S3) la condition interne indicative d'une erreur et pour stocker l'état du système à un moment défini avant que la condition interne indicative d'une erreur ait été détectée (S3) et une trace d'état du système jusqu'au moment où la condition interne indicative d'une erreur a été détectée (S3).

8. Procédé selon l'une des revendications précédentes, comprenant en outre de fournir (S5) des informations sur l'activité du réseau et le fonctionnement interne du système (S) pour analyse.

9. Procédé selon l'une des revendications précédentes, dans lequel l'état enregistré est restauré sur le système (S) avec un code logiciel modifié.

10. Code de programme informatique comprenant des instructions qui, lorsqu'elles sont exécutées par au moins un processeur, amènent l'au moins un processeur à exécuter un procédé selon l'une quelconque des revendications 1 à 9 pour reproduire l'état d'un système (S) et d'un réseau (N) lors de l'apparition d'une erreur.

11. Appareil (20) pour reproduire l'état d'un système (S) et d'un réseau (N) lors de l'apparition d'une erreur, l'appareil (20) comprenant :

- un moyen d'analyse du système (22) configuré pour enregistrer de manière continue (S1) l'état du système (S) pendant une fenêtre temporelle, l'état du système (S) étant enregistré (S1) par un débogueur avec traçage (TD) ;
- un moyen d'analyse de réseau (23) configuré pour générer (S2) un journal de réseau pour le réseau (N) et pour détecter (S3) un état de réseau indicatif d'une erreur ; le journal de réseau pour le réseau (N) étant généré (S2) par un traceur de réseau (NT) ; et
- un dispositif de stockage (24) configuré pour stocker (S4) un état enregistré du système (S) avant l'erreur ainsi qu'un journal des actions système du réseau (N) jusqu'à l'apparition de l'erreur ;

l'appareil étant en outre configuré pour restaurer (S6) l'état enregistré avant l'erreur sur le système (S) et pour rejouer (S7) le trafic réseau enregistré afin de reproduire l'erreur.

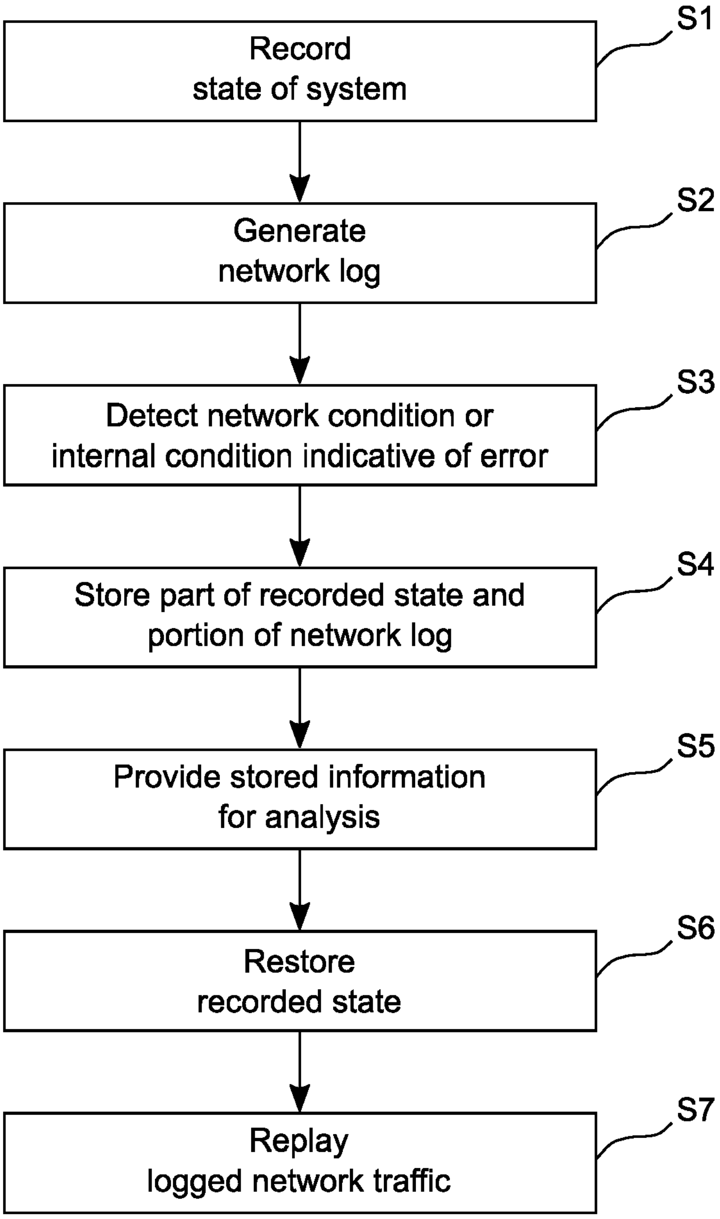
S1
Record state of system
S2
Generate network log
S3
Detect network condition or internal condition indicative of error
S4
Store part of recorded state and portion of network log
S5
Provide stored information for analysis
S6
Restore recorded state
S7
Replay logged network traffic

FIG. 1

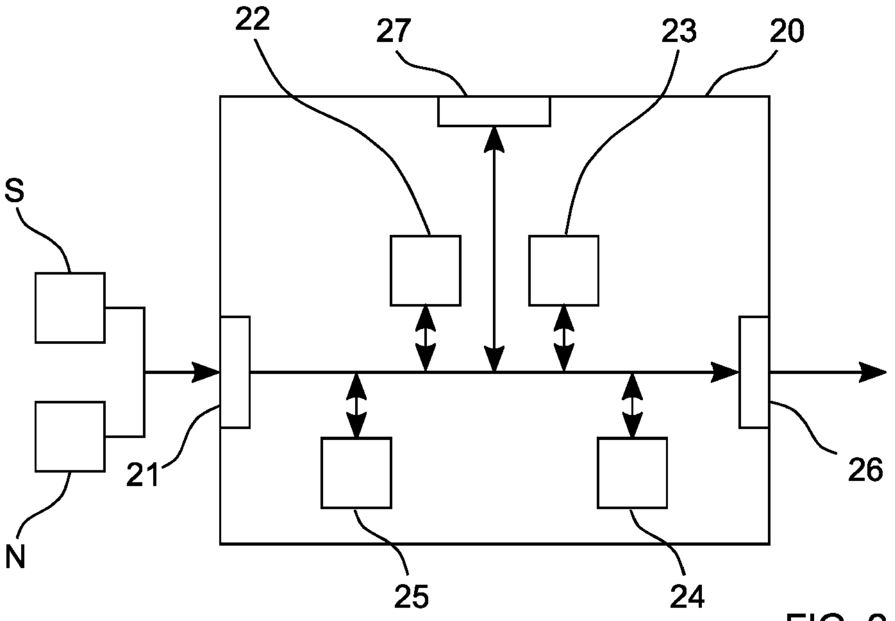

FIG. 2

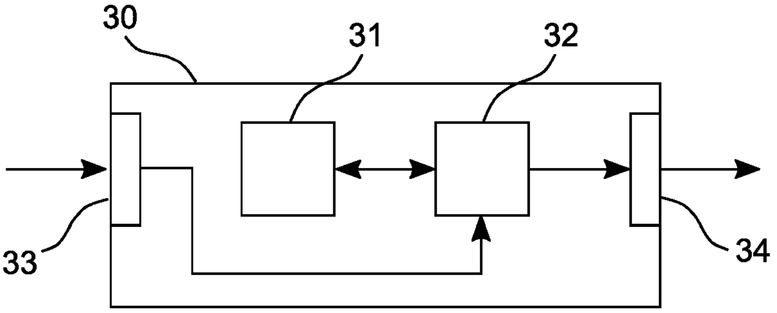

FIG. 3

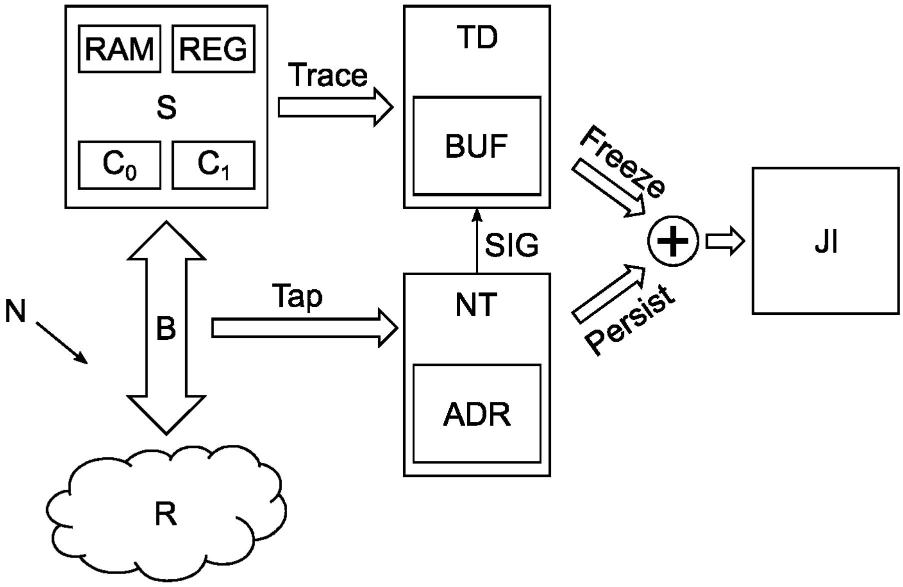
RAM
REG
S
C_0
C_1
Trace
TD
BUF
Freeze
JI
SIG
N
B
Tap
NT
ADR
Persist
R

FIG. 4

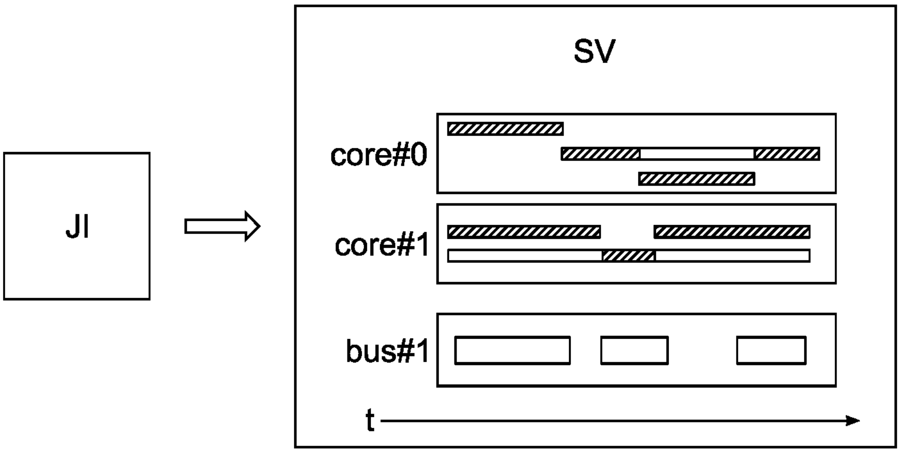
JI
SV
core#0
core#1
bus#1
t

FIG. 5

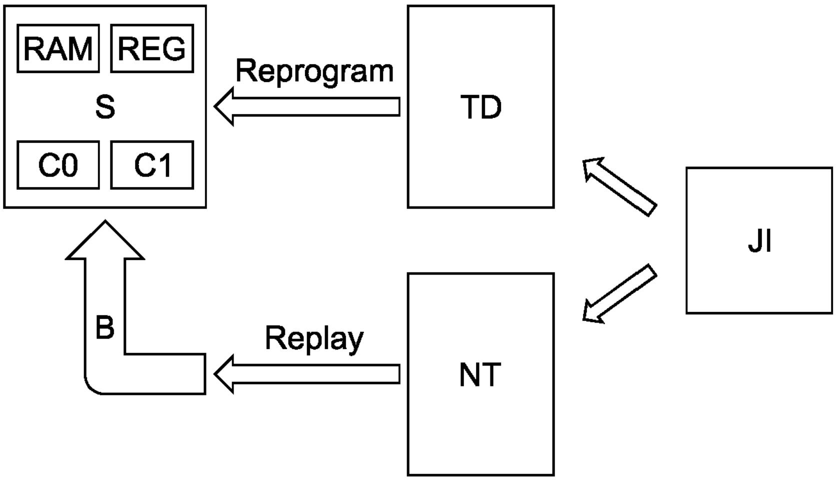
RAM
REG
S
C0
C1
Reprogram
TD
JI
B
Replay
NT

FIG. 6

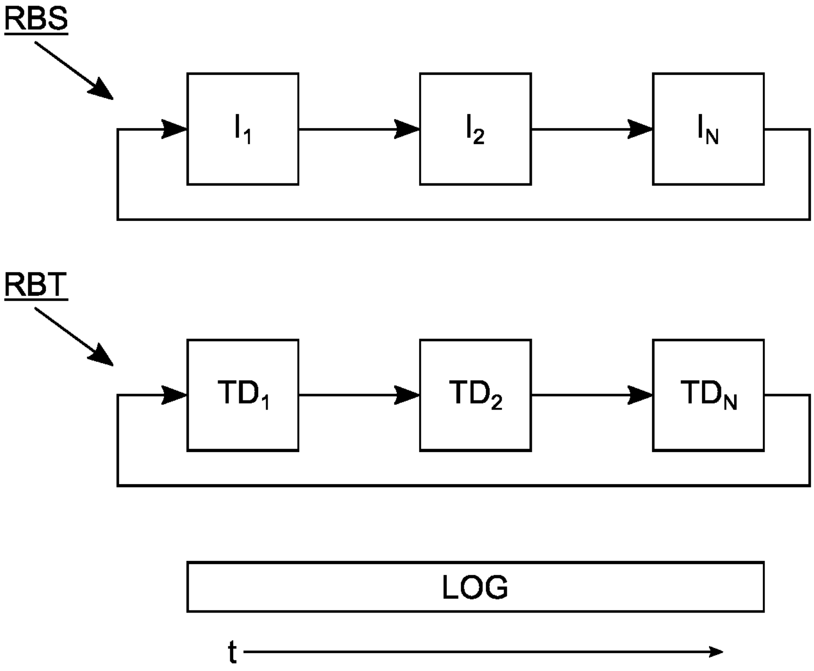
RBS
I_1
I_2
I_N
RBT
TD_1
TD_2
TD_N
LOG
t

FIG. 7

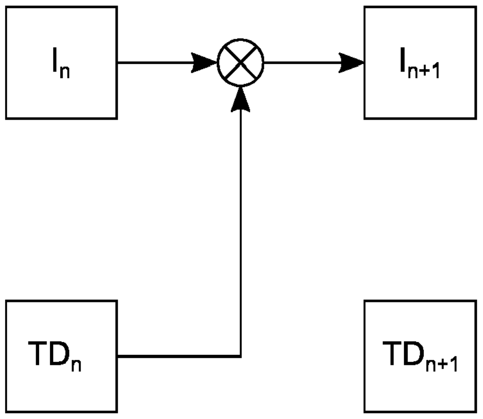

FIG. 8

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- US 20150378870 A1 **[0003]**
- US 20110078666 A1 **[0004]**
- US 8117600 B1 **[0005]**