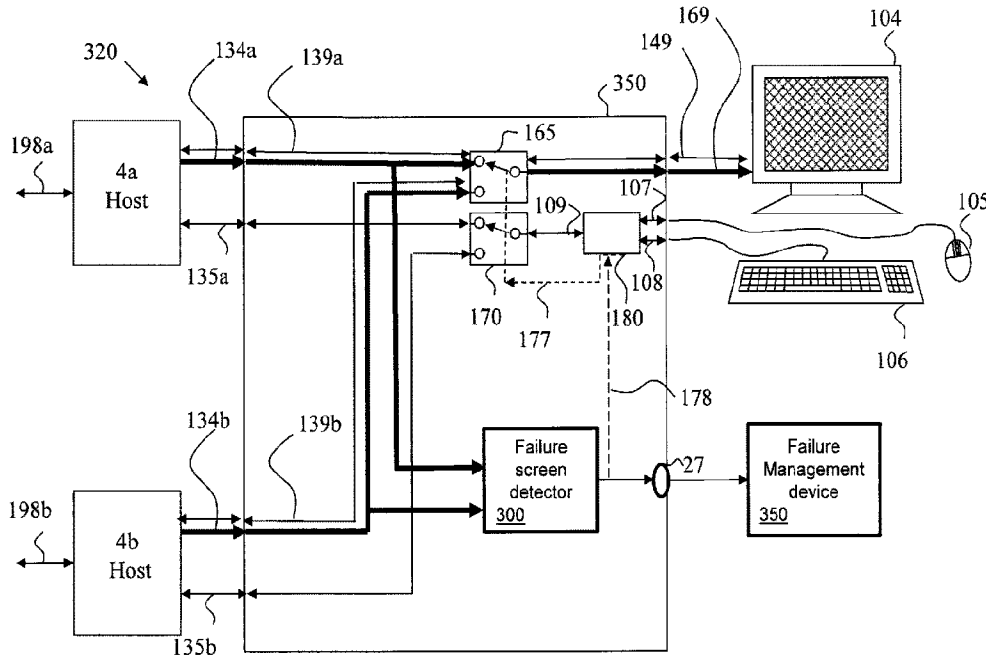




(22) Date de dépôt/Filing Date: 2017/04/13
 (41) Mise à la disp. pub./Open to Public Insp.: 2017/10/14
 (45) Date de délivrance/Issue Date: 2022/10/11
 (30) Priorité/Priority: 2016/04/14 (US62/322,284)

(51) Cl.Int./Int.Cl. *G06F 11/32* (2006.01),
G06F 3/00 (2006.01)
 (72) Inventeur/Inventor:
SOFFER, AVIV, IL
 (73) Propriétaire/Owner:
HIGH SEC LABS LTD., IL
 (74) Agent: TEITELBAUM & BOUEVITCH

(54) Titre : COMMUTATEUR ECRAN-CLAVIER-SOURIS A DETECTION D'ECRAN BLEU DE LA MORT ET FONCTIONS D'AVERTISSEMENT
 (54) Title: KVM HAVING BLUE SCREEN OF DEATH DETECTION AND WARNING FUNCTIONS



(57) Abrégé/Abstract:

A device, apparatus, system and method for determining failure of a computer host among a plurality of hosts. The host failure detection device may be integrated in a KVM apparatus. The device monitors the video output of the plurality of hosts and if identifies a Blue Screen of Death or BIOS failure Black Screen, it issues a warning and logs the details of the discovered failure. The device may attempt to recover the failed host by transmitting emulated keyboard and mouse commands to the failed host.

ABSTRACT

A device, apparatus, system and method for determining failure of a computer host among a plurality of hosts. The host failure detection device may be integrated in a KVM apparatus. The device monitors the video output of the plurality of hosts and if identifies a Blue Screen of Death or BIOS failure Black Screen, it issues a warning and logs the details of the discovered failure. The device may attempt to recover the failed host by transmitting emulated keyboard and mouse commands to the failed host.

KVM HAVING BLUE SCREEN OF DEATH DETECTION AND WARNING FUNCTIONS

FIELD OF THE INVENTION

5

The present invention relates to a device and a method that enables a KVM (Keyboard Video Mouse) switch to detect Blue Screen of Death or similar catastrophic computer failure screens automatically, and generate an automatic warning message to the system administrator once such event is being detected.

10

BACKGROUND OF THE INVENTION

KVM switches are being used to monitor and enable centralized access to multiple servers or computers from a single set of console peripheral devices (keyboard, mouse and display). KVM switches are using by system administrators to monitor and handle computer and server catastrophic failures.

One of the most critical computer or server failure events is the "Blue Screen of Death" (BSoD). The BSoD is an error screen displayed on a Windows computer system after a fatal system error, also known as a system crash, when the operating system reaches a condition where it can no longer operate safely.

Similar critical failure may occur if the computer or server is booting and fails to pass the BIOS stage. In this case known as "black screen" (Bios Black Screen "BBS), typically an administrator manual intervention is needed in order to recover from the failure as additional reset command may cause loop back to the same situation.

In most cases the administrator will be warned by the user once critical failure occurs. Then, the administrator will access the local or remote KVM console to search for the faulty computer or server by switching to the proper channel.

Simple Network Management Protocol (SNMP) is a protocol-based network management system. It is used to manage TCP/IP-based and IPX-based networks.

Information on SNMP can be found in the Internet Request for Comment (RFC) 1157. Microsoft provides an SNMP agent, or client, for Windows NT and Windows 95. Microsoft, however, does not offer any management capabilities. There are third-party companies that offer products specifically designed for SNMP management. SNMP provides the ability to send traps, or notifications, to advise an administrator when one or more conditions have been met. Traps are network packets that contain data relating to a component of the system sending the trap. The data may be statistical in nature or even status related. SNMP traps are alerts generated by agents on a managed device.

10 Chinese patent application CN 104699615A titled "System failure processing method and device", discloses a system failure processing method. The method includes: monitoring whether or not a system running in a virtual machine has a blue screen; if yes, ending a virtual machine thread through an external program; configuring the virtual machine through a preset program to start the system from an external storage device; automatically searching for a memory dump file of system blue screen; after finding the memory dump file, backing up the memory dump file to a specified address.

Chinese patent application CN 103049343 A titled: "Method and device for restoring operating system blue screen", discloses a method comprises the steps of identifying whether bugs recorded in files for restoring system bugs are core files or not and need backup or not through specific fields when the files for restoring system bugs are detected; performing backup of the core files replacing restored files before the bugs needing back up are restored; using the backup core files to cover the restored files causing the blue screen when the restored bugs cause the operating system blue screen.

US patent application 7873498 B2 titled: "Remote hardware inspection system and method", discloses a remote hardware inspection system, including a remote monitor station and a client server. The monitor station includes a hardware inspection unit and a first network interface. The hardware inspection unit is for generating a hardware inspection command which is transmitted to the server by the first network interface.

The server includes a second network interface, an input/output unit, and a managing/controlling unit. The input/output unit has an inspection program. The second network interface receives the hardware inspection command and transmits it to the managing/controlling unit. In response to the hardware inspection command, the managing/controlling unit generates and transmits a trigger signal to the input/output unit to implement the inspection program. Thereby, the inspection program writes data into the target hardware of the server according to the content of the hardware inspection command.

US8769172; titled "Secure KVM device ensuring isolation of host computers"; to Aviv Soffer; discloses apparatuses and systems for operating multiple computers from a single console using a secured KVM device, while preventing information leakage between the computers. The system comprises several hosts connected through a secured KVM device to keyboard and mouse and one or more user displays. Secured KVM enables standard bi-directional communication between Secured KVM and user keyboard and mouse and between hosts peripheral ports and Secured KVM. Secured KVM physically enforces unidirectional data flow from attached keyboard and mouse to attached hosts peripheral ports to avoid potential leakages between hosts.

US20110208963; titled: "Secured kvm system having remote controller-indicator" to Aviv Soffer; discloses a KVM switch capable of providing secure remote extension of KVM control and indication functions.

US20130050084; titled "secure KVM system having multiple emulated EDID functions"; to Soffer Aviv; discloses a KVM (Keyboard Video Mouse) device for operation in high security environments. More specifically, this secure KVM allows flexibility of operation of modern displays while maintaining isolation among the connected hosts.

SUMMARY OF THE EMBODIMENTS

The current invention is generally related to a device, method and system that allow automatic detection of failure or "crash" of a host computer or a server or few host computers or servers from a plurality of such host computers or servers (termed "hosts"

from now on) connected to the fault detection unit. In response to detection of a failure event, the fault detection unit issues a warning signal or a message to the user and/or to computer system administrator. Additionally and optionally, fault detection unit may initiate automatic failure recovery actions. The fault detection unit receives video signals
5 from the plurality of the host and analyze the video signals to determine if, and which host had crashed. The fault detection unit may be used in one of the following ways: 1) As a stand-alone device; 2) Integrated into a KVM switch; and 3) Integrated into other computer networking devices.

As a stand-alone, the fault detection unit, the unit is connected to the video outputs
10 of each of the plurality of monitored hosts to receive video signals. The unit is connected to at least one management device and reports to the management device when a fault event is detected. Optionally, the fault detection unit is connected to the monitored hosts via command channel such as USB channel and is capable to transmit recovery commands to reset, reboot or otherwise attempt to correct or overcome the
15 detected fault. The fault detection unit may probe each of the video signals sequentially. This can be done by using a multiplexor to connect one video signal at a time to a fault detection circuitry. Alternatively, the fault detection circuitry may monitor few or all the video signals in parallel.

The fault detection unit may be integrated into a video switching KVM switch.
20 Video switching KVM switches are known in the art. Thus, in description and drawings some details of the known KVM functionalities may be missing or only briefly explained. In a video switching KVM switch, the user views video from one selected host and interacts with that selected host via a keyboard and mouse. It is the aim of the exemplary embodiment to monitor the non-selected hosts (and optionally also the
25 selected host) for faults. This can be done for example by connecting all the video signals to a fault detection unit integrated into the video switching KVM switch device. Fault alerts can be displayed on the user monitor or using a separate display. A KVM switch is already connected to each of the host via a keyboard and mouse channels (these channels may be united to a single human interface channel such as a
30 composite USB channel). This channel(s) may be used by the user to attempt to recover the crashed host from the fault. Additionally, optionally, or alternatively, this

channel(s) may be used to automatically attempt to recover the affected host from the fault. Optionally, a separate command line, for example a second USB channel is used for the automatic recovery attempt. Optionally, the recovery may be achieved while the user continues to interact with another host or hosts that perform normally. Optionally, once a fault is detected, the KVM selects the affected host and lets the user attempt to recover it.

The fault detection unit may be integrated into a video processing (multi-viewer) KVM switch. Video processing KVM switches are known in the art and. In contrast to a video switching KVM, in a multi-viewer KVM a video processor receives video signals from all the connected host and generate a composite display including images from one or plurality of selected hosts and optionally a generated images that are generated by a video processor within the a multi-viewer KVM. In this exemplary embodiment, the fault detection circuitry is connected to or integrated into the video processor within the multi-viewer KVM.

The fault detection unit may be integrated into a KVM over IP switch. KVM over IP switches are known in the art. In such KVM switches, a remote user interacts with a plurality of hosts that are in a different location and are locally connected to the KVM over IP switch. A KVM over IP switch is similar in its construction and operation to one of the abovementioned KVM switches, but instead of having a local display and human interface devices, the display and human interface devices are at the remote location. The user (or administrator) may interact with a plurality of such KVM over IP switches, and optionally may only infrequently interact with a host, optionally only when the host needs special attention such as upgrading or in case it crushed. It is advantageous to provide a KVM over IP switch with fault detection and alerting capabilities such that the remote user does not have to monitor the health of the multitude of hosts that may be connected to a plurality of KVM over IP switches, but will be automatically informed when a fault occurs. It may be advantageous to further provide a KVM over IP switch with fault recovery and reporting capabilities such that the remote user does not have to monitor and maniacally perform a recovery of each faulty host, but will be automatically informed when a fault occurs and successfully recovered, and alerted when the recovery attempt had failed.

The fault detection unit may be integrated into a Local Rack Access (LRA) Console. LRA consoles are rack-mounted KVM and are known in the art. Such LRA consoles are often used in server farms where large number of hosts are used, thus the advantages of the invention are utilized.

5 The fault detection unit may be integrated into a secure KVM switch. In a secure KVM, absolute data and logic separation is maintained. This properly is of utmost importance for high security installations such as the police, banking, and other high security installations. Preferably, separation among the hosts is achieved using hardware device. When processors are used, it is important to protect the codes against
10 hacking. Generally it may be assumed that any of the hosts may be hacked or infected with a computer virus. Additionally, even the user or the administrator may be suspected in some cases. Unidirectional data flow enforcing circuitries may be added to allow the fault detection circuitry to receive video signals from a plurality of hosts, analyze these signals, inform the user or administrator about the detected fault event,
15 and optionally attempt to recover the affected host, all without compromising the isolation of the hosts.

To shorten the troubleshooting and handling of "Blue Screen" and/or "Black Screen" events, it is desirable to have an automatic detection system will be provided to
20 detect such failure among multiple computers or servers.

In exemplary embodiments, it is further desirable that such detection would be integrated into the KVM to prevent the need to add multiple cabling to each computer or server.

In exemplary embodiments, it t would also be advantageous that this failure
25 detection function would be able to send a warning message once a failure is being detected to the assigned system administrator.

In exemplary embodiments, it t would also be advantageous that this failure detection function would be able to automatically take action to recover from the failure once a failure is being detected.

Exemplary embodiments overcome the disadvantages of prior art by providing a way for the KVM to scan all connected computer video outputs to detect special failure screen such as:

- 5 1. Blue Screen of Death – detection of this event can be done by analyzing the incoming video input and finding the following unique statistical characteristics:
 - a. Image is substantially static (no, or minimal changes from frame to frame). This allows early detection of BSoD. BSoD initially displays memory dump process. In this case the BSoD will change for the first 30 seconds or so to add more lines of memory dump. It may be possible not to wait for the
10 image to become static and shorten the detection of BSoD. It may be enough to check that a preset percentage, for example 90% of the pixels are static after 1 second to conclude that the image is BSoD (together with the other criterions).]
 - b. Large percentage of the display area is Navy Blue color (R=0, G=0, B=130
15 or HEX #000082).
 - c. Other areas are white colored fonts (R=G=B).
2. Black Screen of BIOS (BBS) – detection of this event can be done by analyzing the incoming video input and finding the following unique statistical characteristics:
 - 20 a. Image is substantially static (minimal changes from frame to frame). Similarly to the above, the Black Screen of BIOS may also be a subject of early detection.
 - b. Large percentage of the display area is black color (R=0, G=0, B=0 all <=30).
 - 25 c. Other areas are white colored fonts (R=G=B all >=30).

A table preconfigured inside the KVM device may assign administrator email and phone number for each or all KVM channels. In addition, SNMP trap may be set to enable detection by standard management tools. Once one of the failure screens above

is being detected by the KVM, a special warning message is sent to the assigned administrator.

Such email message may include at least some the following information:

1. The site;
- 5 2. The time of failure;
3. The detected screen type;
4. The name of the server / its exact location in the rack;
5. The failure history of that computer / server (last similar events);

10 When a catastrophic event occurs in a specific server or computer among large number of similar platforms, it is important to understand the potential cause of such event. The history of the specific server or computer may assist in the understanding of the exact cause. Such information may be difficult or even impossible to extract from a server or computer once it failed. In addition to that, similar platforms may experience similar failures in the past. Access to such

15 information may improve the overall server or computer farm up-time and reliability. It should be noted that this statistical data may be collected by other management solutions but in many cases it is difficult to find at the time that the failure occurs.

6. Screen capture of the failure screen; and
- 20 7. Shortcut for remote management over IP of that computer / server.

KVM device may be configured to provide automatic remedy for the detected failure. For example – upon detection of the Blue Screen of Death, the device will use the emulated keyboard function to trigger a key sequence like CTRL+ALT+DEL. Such a sequence, if sent may be used for resetting or rebooting the host computer that

25 generated the BSoD.

Upon detection of the BBS, the device may use the emulated keyboard function to trigger a F8 to command an automatic reboot.

Other command or a sequence of commands may be used as needed.

Additionally or alternatively, the device may use an emulated mouse function to send mouse commands such as clicking on icons or screen positions to cause the computer to recover from its fault state. A combination of emulated mouse function and emulated keyboard function may be used.

5 It should be noted that in the drawings only two hosts are seen in order to reduce figure cluttering. However, generally, the number of hosts connected to a KVM switch or to a fault detection unit may be larger.

10 It is an aspect of the current invention to provide a KVM device for monitoring and operating plurality of coupled computers or servers comprising:

- a plurality of computer interfaces, each capable of receiving video signal from a host computer or server;
- a plurality of video receivers, each coupled to the corresponding one of said computer interfaces to convert video signals received from said computer video interface to common video format;
- 15 - a video processor, capable of receiving at least one video signal in common video format from said video receivers and transmit at least one combined or switched video output to at least one locally connected display;

20 wherein said video processor is further having a video analysis function to enable video statistical recognition of pre-defined failure screens selectable from the list of: Blue Screen of Death; Black Screen of Death; and other predefined failure screens captured by the manufacturer or by the administrator.

In some embodiments the video processor is further coupled to a video transmitter to convert video format received from the video processor it into a standard video display protocol to drive external display.

25

In some embodiments the computer interfaces are further comprising USB device emulators to enable emulation of USB devices selectable from the list of: USB keyboard; USB pointing device; USB audio device; USB floppy drive device; USB Mass storage device; and USB communication class device.

In some embodiments the KVM device further comprises a KVM over IP function to compress and send video output from said video processor to a remote administrator over IP.

5 In some embodiments the KVM device further comprises a Scaler function to scale video information to the format used by the connected display.

In some embodiments the video processor further comprising PiP functionality.

In some embodiments the KVM device further comprises an audible warning device for alerting when a host failure was detected.

10 In some embodiments the KVM device further comprises a warning function, capable of warning a system administrator through one of: email; system message sent to centralized management system; and SMS to the administrator mobile phone.

In some embodiments the warning the administrator is based on a table that predefines the assigned admin to each server.

In some embodiments the warning email includes video capture.

15 In some embodiments the KVM device is capable of generating and maintaining Log detain failure events.

In some embodiments the warning email includes log and history of the failed server.

In some embodiments the KVM device is configured to provide automatic remedy for the detected failure.

20 In some embodiments the automatic remedy for the detected failure comprises upon detection of the Blue Screen of Death, transmitting to the failed host emulated keyboard sequence comprising CTRL+ALT+DEL.

In some embodiments, the KVM behavior does not change once failure detected in a way that may be detectable to the other (non-failed) computers.

25

It is an aspect of the current invention to provide a method for detecting failure of a host comprising: identifying and counting pixels in a first color range in a video frame; identifying and counting pixels in a second color range in said video frame; counting the total number of pixels in said video frame; comparing the number of pixels in said first

color range to said total number of pixel in said video frame; comparing the number of pixels in said second color range to said total number of pixel in said video frame; and determining if the host computer that generated said video frame based on said comparing the number of pixels in said first color range to said total number of pixel in said video frame and said comparing the number of pixels in said second color range to said total number of pixel in said video frame.

In some embodiments the method further comprises field updates to the failure of a host detection algorithm.

In some embodiments the field updates to the failure of a host detection algorithm enables detection of new types of failure screens.

In some embodiments the field updates to the failure of a host detection algorithm enables to skip and avoid false detection of other type of failure screens.

Unless otherwise defined, all technical and scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which this invention belongs. Although methods and materials similar or equivalent to those described herein can be used in the practice or testing of the present invention, suitable methods and materials are described below. In case of conflict, the patent specification, including definitions, will control. In addition, the materials, methods, and examples are illustrative only and not intended to be limiting.

Unless marked as background or art, any information disclosed herein may be viewed as being part of the current invention or its embodiments.

BRIEF DESCRIPTION OF THE OF THE DRAWINGS

Some embodiments of the invention are herein described, by way of example only, with reference to the accompanying drawings. With specific reference now to the drawings in detail, it is stressed that the particulars shown are by way of example and for purposes of illustrative discussion of the preferred embodiments of the present invention only, and are presented in the cause of providing what is believed to be the most useful and readily understood description of the principles and conceptual aspects

of the invention. In this regard, no attempt is made to show structural details of the invention in more detail than is necessary for a fundamental understanding of the invention, the description taken with the drawings making apparent to those skilled in the art how the several forms of the invention may be embodied in practice.

5

In the drawings:

Figure 1A illustrates an example of Windows XP Blue Screen of Death as known in the prior art.

10 **Figure 1B** illustrates an example of Windows Black Screen of Death as known in the prior art.

Figures 2A schematically illustrates a KVM switching system of the prior art.

Figures 2B schematically illustrates a secure KVM switching system of the prior art.

15 **Figures 3A** schematically illustrates a computer system using a KVM switch with a fault detection unit according to an exemplary embodiment of the current invention.

Figures 3B schematically illustrates a computer system using a secure KVM switch with a fault detection unit according to another exemplary embodiment of the current invention.

20 **Figures 3C** schematically illustrates a computer system using a secure KVM switch with a fault detection unit according to an exemplary embodiment of the current invention.

Figures 3D schematically illustrates a secure computer system using a secure KVM switch with a fault detection unit according to another exemplary embodiment of the current invention.

25 **Figure 4A** illustrates a high-level block-diagram of an exemplary embodiment of the present invention having a video processor and local display.

Figure 4B illustrates a high-level block-diagram of yet another high-level block-diagram of another exemplary embodiment of the present invention having a video processor and a remote display, keyboard and mouse over IP.

Figure 4C illustrates a high-level block diagram of a KVM system according to another exemplary embodiment of the current invention having a switching video function.

5 **Figure 4C** illustrates a high-level block diagram of a KVM system according to yet another exemplary embodiment of the current invention having two switching video functions.

Figures 5A illustrates a high-level flow-chart of the failure detection algorithm of an exemplary embodiment of the current invention.

10 **Figures 5B** illustrates a high-level flow-chart of the failure detection algorithm of an exemplary embodiment of the current invention.

DETAILED DESCRIPTION OF THE DRAWINGS

15 Before explaining at least one embodiment of the invention in detail, it is to be understood that the invention is not necessarily limited in its application to the details set forth in the following description or exemplified by the examples. The invention is capable of other embodiments or of being practiced or carried out in various ways.

20 It will be appreciated that certain features of the invention, which are, for clarity, described in the context of separate embodiments, may also be provided in combination in a single embodiment. Conversely, various features of the invention, which are, for brevity, described in the context of a single embodiment, may also be provided separately or in any suitable sub-combination or as suitable in any other described embodiment of the invention. Certain features described in the context of various
25 embodiments are not to be considered essential features of those embodiments, unless the embodiment is inoperative without those elements.

In discussion of the various figures described herein below, like numbers refer to like parts. The drawings are generally not to scale. For clarity, non-essential elements may have been omitted from some of the drawing.

To the extent that the figures illustrate diagrams of the functional blocks of various embodiments, the functional blocks are not necessarily indicative of the division between hardware circuitry. Thus, for example, one or more of the functional blocks (e.g., processors or memories) may be implemented in a single piece of hardware (e.g., a general purpose signal processor or random access memory, or the like) or multiple pieces of hardware. Similarly, the programs may be stand-alone programs, may be incorporated as subroutines in an operating system, may be functions in an installed software package, and the like.

10 **Figure 1A** illustrates an example of Windows Operating System Blue Screen of Death display capture (BSoD) 10. The BSoD is an error screen displayed on a Windows computer system after a fatal system error, also known as a system crash: when the operating system reaches a condition where it can no longer operate safely.

15 In this figure the Navy blue colored background 111 (shown in black color) is visible together with the white fonts 112.

Figure 1B illustrates an example of Windows Operating System BIOS Black Screen display capture 20.

20 In this figure the black colored background 911 is visible together with the white fonts 212.

Figures 2A-B, schematically show KVM system of the art. To avoid cluttering the drawing only essential elements needed to explain the basic operation of these KVM are depicted here. KVM system of the arts may comprise additional elements and functionalities. Such additional details may be found in the prior art publication provided with this application.

Figures 2A schematically illustrates a computer system 120 using a KVM switching apparatus 150 of the prior art.

Hosts or computers 4x ("x" herein and throughout the document refers to any of the letters "a", "b" etc. that indicates identical or similar elements in a plurality of elements. Generally, the number of such elements may be different, and often larger than the number of elements seen in the drawings.) are connected to KVM switch 150 via the corresponding video lines 134x and USB channels 135x. User display 104 is connected to KVM switch 150. User human interfaces such as user keyboard 106 and user mouse 105 are connected to KVM switch 150, for example via USB channels 107 and 108 respectively.

USB controller 180 combines the signals from user keyboard 106 and user mouse 105 into combine USB channel 109. Additionally, USB controller 180 extracts from the signals from user keyboard 106 and user mouse 105 host-selection commands such as keyboard combinations or clicking on specific icons for controlling the host selection switches such as video selector switch 165 and host interface selector switch 170 via command line 177.

Video selector switch 165 connects the selected video input from the selected one of the hosts 4x to be displayed on user display 104. Host interface selector switch 170 switches the signals from user keyboard 106 and user mouse 105 to the selected one of the hosts 4x.

It should be noted that since USB communication is generally bidirectional (as pointed out by the dual headed arrows 154x, 109, 107 and 108, isolation of hosts 4x cannot be guaranteed in this prior art non-secure KVM switch. For example, USB controller 180 may be hacked to be used as a spying device for extracting and storing secret information from high-security host 4a which is connected to high-security network 198a when host 4a is selected. Later, when host 4b which is connected to low security or open Internet 198b is selected, the stored secret information may be transferred from the hacked USB controller 180 to host 4b and from there to the adversary's site over the non-secure network 198b. Additionally, a fake or doctored user human interface 105 or 106 may serve for the same purpose. It is easy to modify a keyboard or a mouse to be used, in addition to their normal operation, as mass storage devices for the purpose of data theft.

A less appreciated is the security vulnerability associated with modern video displays. Generally, display 104 receives video signals via video line 169. However, in order to operate correctly, display data such as Extended Display Identification Data (EDID) is generally and optionally communicated between user display 104 and hosts 2x via bidirectional channel 169 and the corresponding 139x channels. Again, a doctored or hacked display may be used for leaking information among hosts 4x via bidirectional channel 169 and the corresponding 139x channels.

Figures 2B schematically illustrates a secure computer system 129 using a secure KVM switching apparatus 250 of the prior art.

For brevity, only the modifications from figure 2A are discussed herein. It should be noted that a secure KVMs are known in the art and may comprise additional functionalities not discussed herein.

Non-compromised isolation of hosts 4x via the human interface lines is maintained by the following modifications:

USB controller 280, acting as a host emulator in front of user keyboard 106 and user mouse 105, combines the signals from user keyboard 106 and user mouse 105 into combine unidirectional channel 109. Commands unidirectional channel 109 optionally pass through unidirectional data flow enforcement devices 235x to the corresponding USB device emulators 360x which bidirectional communicate with a corresponding hosts 4x. It is clear to see that no data can leak through the human interface channels from one host to the other.

Video data flow enforcement devices 239x inserted in the flow of video signals 134x similarly ensure isolation of hosts 4x with respect to the video channels. EDID (or other display information) is not allowed to be exchanged freely between hosts 4x and user display 104. Instead, display data is exchanged between user display 104 and user display interface unit 210, while each host 4x can exchange display data only with its corresponding display interface units 211x.

Pending applications to the inventor of this invention disclose method of synchronizing data in user display interface unit 210 and data in display interface units

211x. Some embodiments in these application show KVM and secure KVMs having single and multiple user displays and multiple video outputs for each host. It would be easy for a man skilled in the art to extend the current invention to be integrated within KVM having multiple display ports and such KVMs are within the scope of the current invention.

Figures 3A schematically illustrates a computer system 320 using a KVM switch 350 with a failure screen detector 300 according to an exemplary embodiment of the current invention.

The KVM functionality of the embodiment depicted in figure 3A is similar to that depicted in figure 2A and thus, only the added elements aimed at detection, reporting and recovering from failure of a host will be discussed. KVM switch 350 comprises a failure screen detector 300 which receives video signals from hosts 4x. As will be discussed later and specifically in figures 5A-B, failure screen detector 300 is capable to detect at least one or several failure screens such as seen in figures 1A and 1B.

In response to detecting a host failure, failure screen detector 300 may activate via command line 178 the video selector switch 165 and the host interface selector switch 170 to select the failed host. The user then becomes aware that the host had failed by seeing the failure screen on display 104 and can take actions to correct the situation using the user human interfaces such as user keyboard 106 and user mouse 105.

Additionally, optionally, or alternatively, in response to detection a host failure, failure screen detector 300 may take automatic corrective action by commanding the USB controller 180 via command line 178, to execute a pre-defined keyboard and or mouse sequence at the failed host 4x, via USB line 135x, USB switch 170 and USB line 109.

Additionally, optionally, or alternatively, in response to detecting a host failure, Failure screen detector 300 may send a message to an external failure management device 350 connected to optional external port 27. This external port 27 may be connected to other management devices to generate system alarms or warnings. For

example, but not limited to, serial protocol such as RS-232, USB, or Ethernet may be used.

5 It should be noted here that in secure computer system, the design of the USB controller 180 shall be made in such way that will assure that failure of one host would not affect or may not be detectable by the other host(s) through changes in timing or USB switching actions. For example, recovery commands initiated by Failure screen detector 300 cause USB controller 180 to generate recovery keyboard (and/or mouse) command only while host interface selector switch 170 is switched to the failed host.
10 Optionally, the recovery process takes a short time duration, or few short time duration intervals, such that the user ability to continue interacting with the non-failed hosts is only slightly affected or not affected at all.

Optionally, Failure screen detector 300 monitors the recovery process of the failed host. This could be done by analysis of the video signals from the failed host and determining successful recovery. It should be noted that in some cases recovering a
15 host may require a sequence of commands, timed in responses to the recovery progress which may progress at different rate depending on the host and the nature of the failure. In this cases, Failure screen detector 300 may be adopted to recognized the known screens generated by a host during its recovery process and respond accordingly. Optionally and additionally, a log or a message may be generated and
20 optionally sent to the Failure Management device 350 with details of the successful recovery. Optionally and additionally the number of recovery attempts may be limited, and log or a message may be generated and optionally sent to the Failure Management device 350 if recovery attempts have not been successful.

25

Figures 3B schematically illustrates a computer system using a KVM switch with a fault detection unit according to another exemplary embodiment of the current invention.

The KVM functionality of the embodiment depicted in figure 3B is similar to that depicted in figure 3A and thus, only the added elements will be discussed.

In order to reduce cabling, signals 109 from user interface devices 105 and 106 are combined with host-recovery signals from the appropriate device emulator 360x in combiner such as USB HUB 361x which is connected to the failed host 4x.

Optionally, the user can continue to interact and view any of the host.

5 Optionally the user can select the failed host. In this case, user commands may take priority over the recovery command that automatically generated by failure screen detector 300. Optionally, the user can halt or cancel the automatic recovery attempts.

10 It should be noted that device emulators 360x and USB HUBs 361x may also be used to "keep alive" the USB bidirectional connection between the KVM or failure detection device and the non-selected host. Without such devices, any time a host 4x is selected, USB connection has to be established or re-established. This time delay may be annoying when a user is actively interacting with a plurality of hosts 4x and frequently changes the selected host. However, when the KVM or failure detection device is primarily used for monitoring and failure detection and recovery (manual or
15 automatic), such device emulators 360x and USB HUBs 361x may be eliminated for reducing cost and complexity.

Figures 3C schematically illustrates a secure computer system 329 using a secure KVM switch 352 with a fault detection unit 359 according to an exemplary embodiment
20 of the current invention.

 The embodiment depicted in this figure combines the reduce cabling of figure 3B with the host isolation and security of figure 2B, and thus, only the added or modified elements will be discussed.

25 In this figure, failure screen detector 359 is seen integrated with device emulators and is capable of transmitting host recovery commands, optionally via unidirectional data flow enforce device 295x to device emulator 360x. Device emulator 360x emulates keyboard and/or mouse signals in response to user commands 209 or automatic recovery commands from screen failure detector 359.

30 It should be noted that screen failure detector 359 may further affect the selection switches 165 and 170 as seen in figure 3B. It should be noted that device emulators

360x separated from screen failure detector 359 may be used as seen in figure 3B. It should be noted that unidirectional data flow enforce device 295x may be integrated in one of: device emulators 360x; screen failure detector 359 and/or device emulators 360x.

5 It should be noted that a dedicated screen failure detector may be connected to each host 4x. A drawing showing this embodiment was omitted for its complexity, but can be easily constructed by a man skilled in the art. In this case, isolating of hosts is ensure in the failure detection and recovery channels may be maintained without some of the unidirectional data flow enforce devices.

10 To save cost, a video scanning switch 399 may be used to connect one video signal from one host to the Failure screen detector 359 at a time. Video scanning switch 399 may be controlled by Failure screen detector 359 via command line 398. This arrangement allows scanning the video signals from the plurality of hosts, and recovering any of the hosts regardless of the selection of switches 165 and 170. Thus,
15 user interaction (viewing and commanding) of the hosts 4x is not affected or impeded by the video signal scanning, except when trying to send both user's commands and automatic recovery commands to same failed host. Optionally, user selection of failed host halts the automatic recovery process. It should be noted here that such arrangement may require active video splitters 204x to split the incoming video signals
20 from computers 4x to video switch 399. As two video splitters may add cost to the KVM 352, it is possible to eliminate one video splitter 204x and the video switch 399 by coupling the Failure screen detector 359 to the video switch 165 output through a single splitter. Such arrangement is shown in figure 3D below. While such change may not provide a significant cost saving in 2-Port KVM shown here, it may provide much more
25 cost savings in 8, 16 or 32 port KVMs.

Figures 3D schematically illustrates a secure computer system 389 using a secure KVM switch 381 with a fault detection unit 359 according to another exemplary embodiment of the current invention.

The embodiment depicted in this figure combines the reduce cabling of figure 3B with the host isolation and security of figure 2B, and thus, only the added or modified elements will be discussed.

To reduce cost and simplify the system, Failure screen detector 359 is connected
5 behind the video switch 165 to get one selected video signal 205. Controller 280 is adopted to perform scanning the video signals from the plurality of hosts 4x. It should be noted that “splitting” off the video signal as shown in figures 3A and 3B generally requires to add an active splitter which adds cost and complexity.

Video signal 205 is switched video from video splitter 204 that was added behind,
10 or integrated as an additional output to the video switch 165 output. Failure screen detector 359 is having a single video input in this case to receive video signal 205. Control of the video switch 165 is done through command line 206 and controller 280. Since controller 280 is also the keyboard and mouse host emulators same command line 206 may be used to insert any sequence of desired response keyboard or mouse
15 commands to device emulators 360x.

While this arrangement saves video splitters and video switch compared to the arrangement shown in figure 3C above, it is lacking the option that the user will work on one computer while the Failure screen detector 359 will continue scanning computers at the background. Decoupling of the user operation and the Failure screen detection may
20 be essential in large KVMs to assure that these two functions would not negatively affect one another. Optionally, scanning of video switch 165 may be paused when user input via keyboard 106 or mouse 105 is detected and resumes after a preset time of user’s inactivity.

In an exemplary of the embodiment seen in figure 3D, the user’s display 104
25 shows only the video signal of the user selected host. During the brief host analysis periods, the video signal from the currently scanned host is sent to the Failure screen detector 359, and the image on user display 104 freezes or get blank, but the video image from the user selected host is not replaced with image from the currently scanned host. Thus, the user is not subject to uncertainty as to the source of the image
30 seen on display 104.

It should be note that the data security is an added feature and is not essential component of the current invention. Currently, secure KVMs are not in general use in data-centers where the invention may be in use.

5 **Figures 4A-C** illustrates a high-level block diagram of a KVM systems according to an exemplary embodiments of the current invention. In figures 4A-C, more details of the video switching and the failure screen detection are seen and details of the KVM functionalities which are known from previous figures and the prior are were omitted to reduce cluttering of the drawings.

10

Figure 4A illustrates a high-level block diagram of a combiner or multi-viewer KVM system 30 according to an exemplary embodiment of the current invention.

15 The combiner or multi-viewer KVM device 32 of this exemplary embodiment of the current invention is depicted here as having only two channels for supporting two hosts 4a and 4b, but any number of channels may be added to support larger number of computers / servers 4x. Number of channels in a typical KVM is 8, 16 or 32. This combiner or multi-viewer KVM 32 is having multiple DVI / HDMI / VGA or DisplayPort receivers 8x coupled to computer or server 4x through video plug 6x and video cable 5x
20 respectively.

In this discussion, the letter “x” after an element number (e.g. “8x” and “4x” above) indicates any element from a plurality of the similar or identical elements.

25 Video receiver 8x is used to convert the incoming video to a digital format 9x such as 24 bit per pixel LCD parallel bus to support communications with the video processor (which can be realized in the form of a processor, FPGA or ASIC) 11. Video processor 11 may have multiple video receivers 8x integrated internally.

Video processor 11 may have external frame buffer memory 12 or internal volatile memory function to store incoming and outgoing video frames. Memory function may be integrated inside the video processor component.

Failure screen detection function 23 programmed into the video processor 11 to analyze video statistical features of incoming video signals in order to effectively detect the various failure characteristics of BSoD, BBS, and other failure screens. Failure screen detection function 23 is optionally connected to optional external port 27 for transmitting alert message when failure is detected or suspected. This port may be connected to other management devices to generate system alarms or warnings. For example, but not limited to, serial protocol such as RS-232, USB, or Ethernet may be used.

As a stand-alone host monitoring-only unit, device 32 of this exemplary embodiment of the current invention may substantially comprise the above mentioned components only.

When used within a KVM switch, Video processor 11 may further have firmware functions to support prior-art user functions that support KVM switching activity such as:

- a) Video source switching – for displaying on a user display a selected one of the video signals from the selected one of hosts 4a;
- b) OSD (On Screen Display) - to add text and icons to the displayed image;
- c) PiP (Picture in Picture) - to show multiple images from a plurality of hosts 4x on a single user display (display screen 104, not seen in this figure). PiP function may be used for automatically select a failure screen or video signal from a host suspected to fail in order to alert the user that an unselected host had failed; and
- d) Scaler - to enable video scale up and down as needed to display one or multiple sources on the user display.

Optionally, resulted video output from video processor is coupled through video transmitter 16 to the video output port 24. This port may be connected to an external display such as LCD monitor or built-in LCD tray module (For example, the built-in LCD tray module “Avocent 18.5” Local Rack Access (LRA) Consoles” from Emerson Network Power, with Global Headquarters at 1050 Dearborn Drive P.O. Box 29186 Columbus, Ohio 43229, or similar devices).

Figure 4B illustrates a high-level block diagram of another combiner or multi-viewer KVM system 40 according to an exemplary embodiment of the current invention similar to the KVM system 30 of figure 4A above.

5 The combiner or multi-viewer KVM device 42 is having same local video display port 24 but further having KVM over IP function 33. The KVM over IP function is coupled to the video processor 11 to receive the resulted video output 41 selected by the administrator. Typically, the administrator is remotely located and controls system 40 using an IP terminal.

10 KVM over IP function 33 is also coupled through serial line 25 to the Failure screen detection function 23 to receive signals indicative of detection and recognition of the failure conditions in each connected hosts 4x. Once failure condition in one of the connected hosts 4x is detected by the Failure screen detection function 23, it notifies the KVM over IP function 33 to enable remote administrator alerting through means
15 such as:

- a) Email;
- b) System message sent to centralized management system; and
- c) SMS to the administrator mobile phone.

20 KVM over IP function 33 optionally further stores a log of detected events on non-volatile memory integrated internally or connected externally (not shown in figure 4B). KVM over IP function 33 is further coupled to a LAN (Local Area Network) interface 28 to enable the device 42 to communicate with local or remote administrator through a standard web-browser. Not that in this embodiment, external port 27 may optionally be missing.

25 Optional USB device emulators 13x are coupled through USB cables 7x having USB Type-A plugs 3x to their respective computer / server 4x. These emulators 13x can emulate one or more of the following USB device functions:

- a) USB keyboard;
- b) USB pointing device;

- c) USB audio device;
- d) USB external drive device;
- e) USB Mass storage device; and
- f) USB communication class device.

5 Device emulators 13x are coupled through bidirectional serial lines 15x to the KVM over IP function 33 to enable remote operation of the selected host 4x.

KVM over IP function 33 may be used to enable remote connection of the same device listed above. For example – keyboard and mouse may be used at the remote administrator station, to communicate with the KVM over IP function 33 through LAN or
10 WAN. This setup allows remote administrator keyboard and mouse commands to be received by the KVM over IP function 33 and then sent to the selected host 4x through serial line 15x, device emulator 13x, USB cable 7x and USB plug 3x. Similarly, a USB mass storage device, or other data storage, may be connected at the remote location and communicate via the same path (over IP) to the selected computer 4x to enable
15 files copy to and from the computer. For example, such data may appear to host 4x as an external drive from which it can be booted or upgrade its software or operation system in order to recover from a failure, or to prevent future failures.

Figure 4C illustrates a high-level block diagram of yet another KVM system 90
20 according to an exemplary embodiment of the current invention similar to the KVM system 40 of figure 4A above having video switching function 92.

In this embodiment of the current invention, device 91 is not capable of supporting simultaneous display of multiple channels. In this device 91, input video signals from computers 4a and 4b is switched by video multiplexer or switch function 92. This switch
25 is controlled by microcontroller function 97 based on user commands.

In an exemplary embodiment of the invention, when the user is not working with the device 91, the microcontroller 97 initiates automatic scanning of the channels (that is: switch 92 periodically connects to one host 4x at a time). This is done by changing command line 98 at fixed or pseudo-random intervals (for example each 0.5 seconds).

Video multiplexer or switch function 92 will then switch between the different computer sources 4a and 4b etc. at time interval. Video output of switch 92 is coupled to video splitter circuitry 93 to enable video output to the user display through lines 95 and console port 24 and at the same time it will feed same video through line 94 to video failure detection block 23. Once video failure detection block identifies a failure, it will trigger signal on line 96 to the microcontroller function 97. Microcontroller function 97 will then initiate an alarm signal through port 27 to other devices or management consoles. Port 27 may be serial port, USB, Ethernet or any other communication protocol.

10

Figure 4D illustrates a high-level block diagram of yet another KVM system 490 according to an exemplary embodiment of the current invention similar to the KVM system 90 of figure 4B above having two video switching functions 92 and 492.

In this embodiment of the current invention, device 491 is not capable of supporting simultaneous display of multiple channels, but it can consciously monitor all (selected and unselected) hosts 4x. In this device 491, input video signals from computers 4a and 4b is switched by video multiplexer or switch function 92. This switch 92 is controlled by microcontroller function 97 based on user commands.

In an exemplary embodiment of the invention, while the user is working with the device 91, the microcontroller 97 initiates automatic scanning of the channels (that is: switch 492 periodically connects to one host 4x at a time). This is done by changing command line 498 at fixed or pseudo-random intervals (for example each 0.5 seconds). Video multiplexer or switch function 492, which is connected to video outputs 5x of hosts 4x will then switch between the different computer sources 4a and 4b etc. at time interval. Video output of switch 492 will feed its selected video through line 494 to video failure detection block 23. Once video failure detection block identifies a failure, it will trigger signal on line 96 to the microcontroller function 97. Microcontroller function 97 will then initiate an alarm signal through port 27 to other devices or management consoles. Port 27 may be serial port, USB, Ethernet or any other communication protocol.

30

Additionally and optionally, microcontroller function 97 may force switch 92 to select the failed host, or toggle repeatedly between the user select host and the fail hosts. Optionally, scanning and searching for failed host or plurality of failed hosts while the user switch among hosts, and irrespectively to the user selected host, and while a host or a plurality of host are in failed state.

Figures 5A-B illustrate a high-level flow-chart diagram of the detection algorithm used by the video processing function of the exemplary embodiment of the current invention.

It should be noted that blocks in figures 5A-B may be realized in hardware or software or combinations thereof.

Figures 5A illustrates the frame filtering stage of the detection algorithm used by the video processing function of the exemplary embodiment of the current invention.

The failure detection system 50 is aimed to detect specific failure screens such as BSoD and or BBS from the received video input data pixel 52. Upon positive failure detection it produces an alarm or raising an indication output signal.

Note that the detection system 50 shown in figures 5A and 5B is targeted to handle a single received video input channel. To support multiple computers / servers (in KVM for example) the following changes may be implemented:

- a) Multiple instances of this system may be implemented to enable simultaneous detection of multiple sources (see figure 4 above); or
- b) Video switching circuitry may be added before the detection logic so that it will switch between the multiple channels on a timely basis (for example each 0.5 seconds, see figure 4B above).

The video input signals into the failure detection sub-system contain the following standard digital video components:

- Video pixels data 52

- DE (Data Enable) signal 53
- V-Sync (Vertical synchronization) signal 64

For detecting a failure screen the received active video pixels (received only during the DE 53 active period) are analyzed on the first stage, which involves Pixel Color
5 Filters 55, 56 and 57.

Each such a Pixel Color Filter is pre-configured to pass only pixels which are belong to a predefined color range. For this purpose, three Pixel Color Filters are involved, for analyzing pixels in the color ranges: Black 55, White 56 and Blue-Navy 57. The predefined values for the color range for each Pixel Color Filter can be configured
10 individually from the controlling microcontroller (MCU) 82.

Each filter (55, 56, 57) acts on the all the pixel in the frame 52 and set the pixel value to "1" if the pixel color is within the range specified to the filter and "0" if the pixel color is outside the range specified to the filter in the corresponding filtered frame output (59,
61, 62).

15 An indication signal generated as an output (59, 61 and 62) from each Pixel Color Filter (55, 56 and 57 respectively) is sensed on the second stage by successive Frame Pixels Counters (65, 66 and 67 respectively). Each dedicated Frame Pixels Counter counts the overall number of pixels in a specific color range during a complete frame.

In addition to the Frame Pixels Counters which count only the pixels belong to a
20 specific color range as indicated from the Pixel Color Filter, the total number of pixels in a frame are counted by the Total Frame Pixels Counter 68.

The V-Sync input signal 54 is analyzed by the V-Sync Edge Detector module which is sensitive to the rising edge of the vertical synchronization signal. In this case a new frame indication signal is generated. Alternatively the V-Sync Edge Detector 58 can be
25 implemented with a sensitive to the falling edge of the V-Sync instead of the rising edge.

The new frame indication signal is used by the Frame Pixels Counters (65, 66 and 67) and by the Total Frame Pixels Counter 58 for clearing and initializing the pixels

counts to be prepared for counting the successive incoming frame through clear signal 63. Frame pixel counters 65, 66, 67 and 68 outputs are 69a, 69b, 69c and 69d respectively.

The abovementioned filtering technique is but an example for pixel color analysis. Other alternative method may be used. For example, pixel's color value may be sequentially analyzed one after the other:

- a) Total pixel counter 68 will be incremented for each analyzed pixel;
- b) Black pixel counter 65 will be incremented if the color value of the analyzed pixel is in the predefined black color range;
- 10 c) White pixel counter 66 will be incremented if the color value of the analyzed pixel is in the predefined white color range; and
- d) Blue pixel counter 67 will be incremented if the color value of the analyzed pixel is in the predefined blue color range.

15 **Figures 5B** illustrates the fault determination stage of the detection algorithm used by the video processing function of the exemplary embodiment of the current invention.

As soon as a complete frame is analyzed all the produced counts are used for calculating the failure detection by the Failure Frame Detector module 72. In this module 72 the counts of the selected colors (Black 65 = A, White 66 = B and Blue-Navy 67 = C) and the total numbers of frame pixels (Total 69 = D) are examined in order to fill the following conditions in order to detect failure screen:

- Is the number of the Blue-Navy pixels in a frame is high enough to go over a high threshold calculated from the total number of pixel in a frame (73a)? Output of this block is 74a.
- 25 - Is the number of the Black pixels in a frame is high enough to go over a high threshold calculated from the total number of pixel in a frame (73b)? Output of this block is 74b.

- Is the number of the White pixels in a frame is low enough to go under a low threshold calculated from the total number of pixel in a frame (73c)? Output of this block is 74c.
- 5 - Is the number of the Black pixels plus the number of the White pixels is high enough to be above a high threshold calculated from the total number of pixel in a frame (73d)? This indication is used to conclude that there are no, or only few other received color other than the Black and White colors. Output of this block is 74d.
- 10 - Is the number of the Blue-Navy pixels plus the number of the White pixels is high enough to be above a high threshold calculated from the total number of pixel in a frame (73e)? This indication is used to conclude that there are no other received color other than the Blue-Navy and White colors. Output of this block is 74e.

15 The calculated high and low thresholds are calculated as predefined percentages of the frame total pixels count 69d. The predefined values for each threshold can be configured individually from the controlling MCU 82.

A logic block named Failure Frame Detection Table 75 detects for fulfillment of some of the conditions in order to conclude if the received frame is either a failure frame (and identify its exact type) or a normal (not a failure) frame.

20 The Failure Frames Counter 83 counts for consistent detected failed frames 76 and initialize the count when a normal frame is detected. In this manner only a successive input of failure frames are used to detect a stable failure state. The count output 84 generated by the Failure Frames Counter 83 is examined by the comparator 85 which compares the number of the successive failure frames and initiates a fault alert when
25 successive failure frames is over a minimum threshold, indicating on a stable failure state. The predefined value for the threshold can be configured from the controlling MCU 82.

The indication on a stable failure state is sensed by a Failure Alert Generator module 86 which produce an alarm and/or raises an indication signal on the detection of

the failure state through output line 87. On a specific implementation case this module can be implemented as a simple pass-through of the previous comparator result.

5 The configuration MCU 82 block included in the detection diagram is not restricted to be a dedicated MCU integrated in each failure detection for each separate video input channel.

This MCU function may be combined with the controller function 97 shown in figure 4B above.

10 The output of the failure detection diagram 87 is an alarm indication or another indication which is used to trigger a successive alarm/message report function, which is outside the scope of the described system diagram.

15 Optionally, a frame stability is also analyzed. For example, each pixel color value in a currently analyzed frame is subtracted from the color value of the corresponding pixel in the previously analyzed frame. After the subtraction, the number of pixels with subtracted value not equal zero (changed pixels) is determined. If the number of changed pixels is above a preset percentage of the total pixels in the frame, the image is not static.

Optionally, testing for static image is performed only if a plurality of faulty frames were detected. Alternatively, pixel color analysis image is performed only if a plurality of static frames were detected.

20 Optionally a template failed screen or plurality of template failed screens are saved and are compared to the frame from a host to determine if the host is displaying one of the failed screen templates. Such comparison may be done by subtracting the template from the host frame and determining the percentage of non-zero pixels.

25 Optionally a template of legitimate screen or plurality of legitimate templates screens, which are similar to failed screens but indicative of normal operation of a host are saved and are compared to the frame from a host to determine if the host is working properly while displaying screen similar to a failed screen. Such comparison may be done by subtracting the template from the host frame and determining the percentage of non-zero pixels.

As used herein, the term "computer" or "module" may include any processor-based or microprocessor-based system including systems using microcontrollers, reduced instruction set computers (RISC), application specific integrated circuits (ASICs), logic circuits, and any other circuit or processor capable of executing the functions described herein. The above examples are exemplary only, and are thus not intended to limit in any way the definition and/or meaning of the term "computer".

It is to be understood that the above description is intended to be illustrative, and not restrictive. For example, the above-described embodiments (and/or aspects thereof) may be used in combination with each other. In addition, many modifications may be made to adapt a particular situation or material to the teachings of the various embodiments of the invention without departing from their scope. While the dimensions and types of materials described herein are intended to define the parameters of the various embodiments of the invention, the embodiments are by no means limiting and are exemplary embodiments. Many other embodiments will be apparent to those of skill in the art upon reviewing the above description. The scope of the various embodiments of the invention should, therefore, be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled. In the appended claims, the terms "including" and "in which" are used as the plain-English equivalents of the respective terms "comprising" and "wherein." Moreover, in the following claims, the terms "first," "second," and "third," etc. are used merely as labels, and are not intended to impose numerical requirements on their objects.

Further, the limitations of the following claims are not written in means-plus-function format and are not intended to be interpreted based on 35 U.S.C. § 112, sixth paragraph, unless and until such claim limitations expressly use the phrase "means for" followed by a statement of function void of further structure.

This written description uses examples to disclose the various embodiments of the invention, including the best mode, and also to enable any person skilled in the art to practice the various embodiments of the invention, including making and using any devices or systems and performing any incorporated methods. The patentable scope of the various embodiments of the invention is defined by the claims, and may include other examples that occur to those skilled in the art. Such other examples are intended

to be within the scope of the claims if the examples have structural elements that do not differ from the literal language of the claims, or if the examples include equivalent structural elements with insubstantial differences from the literal languages of the claims.

- 5 Although the invention has been described in conjunction with specific embodiments thereof, it is evident that many alternatives, modifications and variations will be apparent to those skilled in the art. Accordingly, it is intended to embrace all such alternatives, modifications and variations that fall within the spirit and broad scope of the appended claims.

10

CLAIMS

1. A Keyboard Video Mouse (KVM) device for monitoring and operating a plurality of coupled computers or servers, comprising:
- 5 a plurality of computer interfaces, each configured for receiving video signal from a corresponding host computer or server;
- a plurality of video receivers, each coupled to the corresponding one of said computer interfaces;
- 10 a video processor, configured for receiving at least one video signal and transmit at least one video output to at least one locally connected display,
- wherein said video processor has a video analysis function, to detect a failure of a host or server based on analysis of said at least one video signal by video recognition of pre-defined failure screens selectable from a list of: Blue Screen of Death, Black Screen of Death, and at least one other predefined failure screen,
- 15 captured by a manufacturer or by an administrator;
- wherein said video recognition comprises at least: capturing a video frame from said video signal: identifying and counting pixels in a first color range in said video frame; identifying and counting pixels in a second color range in said video frame; counting the total number of pixels in said video frame; comparing the
- 20 number of pixels in said first color range to said total number of pixel in said video frame; comparing the number of pixels in said second color range to said total number of pixel in said video frame; and determining if the host computer that generated said video frame has failed, based on said comparing the number of pixels in said first color range to said total number of pixel in said video frame
- 25 and said comparing the number of pixels in said second color range to said total number of pixel in said video frame, wherein a failure is detected when at least the number of pixels in said first color range and the number of pixels in said second color range are higher than a predefined threshold calculated from said total number of pixels in said video frame; and a video scanning switch to

automatically connect one of said plurality of computer interfaces at a time to said video processor.

2. The KVM device of claim 1 characterized in that at least one of said computer
5 interfaces further comprises a USB device emulator to enable emulation of USB devices selectable from a list of:

- a) USB keyboard;
- b) USB pointing device;
- c) USB audio device;
- 10 d) USB floppy drive device;
- e) USB Mass storage device; and
- f) USB communication class device.

3. The KVM device according to claim 1 or 2 , further comprising a KVM over IP function
15 to compress and send video output from said video processor to a remote administrator over IP.

4. The KVM device according to any one of claims 1 to 3, characterized in that said
video processor further comprises PiP functionality.

20 5. The KVM device according to any one of claims 1 to 4, characterized by being configured for generating and maintaining Log detailing failure events.

6. The KVM device according to any one of claims 1 to 5, characterized by a warning
25 function, configured for warning a system administrator through at least one of:

- a) email;
- b) system message sent to centralized management system;

Patent

- c) SMS to the administrator mobile phone; and
- d) audible warning.

5 7. The KVM device according to claim 6, characterized in that said warning the system administrator is based on a table that predefines the assigned at least one administrator to each coupled host computer or server.

10 8. The KVM device of claim 6 or 7, characterized in that said email includes video capture.

10 9. The KVM device according to any one of claims 6 to 8, characterized in that said email includes log and history of a failed server.

15 10. The KVM device according to any one of claims 1 to 9, characterized in that it is configured to provide automatic remedy for a detected failure.

20 11. The KVM device according to claim 10, characterized in that said automatic remedy for the detected failure comprises: upon detection of the Blue Screen of Death, transmitting to the failed host computer or server emulated keyboard sequence comprising CTRL+ALT+DEL.

25 12. The KVM device according to any one of claims 1 to 11, characterized in that behavior of the KVM device does not change once failure detected in a way that may be detectable to other, non-failed, computers, hosts, or servers.

13. The KVM device according to any one of claims 1 to 12, wherein said video processor is further coupled to a video transmitter to convert video format received from said video processor into a standard video display protocol to drive an external display.

14. The KVM device according to any one of claims 1 to 13, further comprising a Scaler function to scale video information to the format used by the connected display.

5 15. The KVM device according to any one of claims 1 to 14, wherein said video scanning switch is independent from switches used to switch the keyboard video and mouse of the KVM device.

16. A method for detecting failure of a host computer or a server comprising: connecting
10 to a plurality of video outputs of each of a plurality of monitored host computers or servers; receiving a plurality of video signals, each from a host computer or a server of the plurality of monitored host computers or servers; automatically selecting at least one video frame from one host computer or a server at a time; identifying and counting pixels in a first color range in said video frame; identifying and counting pixels in a
15 second color range in said video frame; counting the total number of pixels in said video frame; comparing the number of pixels in said first color range to said total number of pixel in said video frame; comparing the number of pixels in said second color range to said total number of pixel in said video frame; and determining if the host computer that generated said video frame has failed, based on said comparing the number of pixels in
20 said first color range to said total number of pixel in said video frame and said comparing the number of pixels in said second color range to said total number of pixel in said video frame, wherein: a failure is detected when at least the number of pixels in said first color range and the number of pixels in said second color range are higher than a predefined threshold calculated from said total number of pixels in said video
25 frame; and the determining recognizes said video frame to be a display of a pre-defined screen selectable from a list of: Blue Screen of Death, Black Screen of Death, and at least one other predefined failure screen, captured by a manufacturer or by an administrator.

17. The method of claim 16, further comprising field updating at least one parameter used in said comparing the number of pixels in said first color range to said total number of pixel in said video frame or in said comparing the number of pixels in said second color range to said total number of pixel in said video frame in order to detect a failure of
5 a host computer or a server.

18. The method of claim 17, wherein said field updating of said at least one parameter to the failure of a host computer or a server enables detection of new types of failure screens.
10

19. The method of claim 17, wherein said field updating of said at least one parameter to the failure of a host computer or a server enables to skip and avoid false detection of non-failure screens.
15

20 →

```

TRAP 000d  ERACD=0900  ERACC=0093  ERLIM=0000ffff
EAX=f09af48  EBX=fff2a7b8  ECX=00000160  EDX=7b32ef1c
ESI=00000000  EDI=fd22d21  EHP=00004eb0  FLG=00003286
CS:EIP=0160:ffffb23d  CSACC=c0f3  CSLIM=ffffffffff
SS:ESP=0030:00004e98  SSACC=1097  SSLIM=00003fff
DS=0150  DSACC=c0f3  DSLIM=ffffffffff  CR0=c001001b
ES=015B  ESACC=c0f3  ESLIM=ffffffffff  CR2=ffc6209b
FS=0133  FSACC=00f3  FSLIM=00004a55
GS=0130  GSACC=00f3  GSLIM=00004a55

The system detected an internal processing
error at location #0160:fff5cc7f - 000d:9c7f.
60000, 9084
048600b4
Internal revision 6.617, 94/01/28

The system is stopped. Record the location number
of the error and contact your service representative.

```

212 →
911 →

Figure 1B (Prior Art)

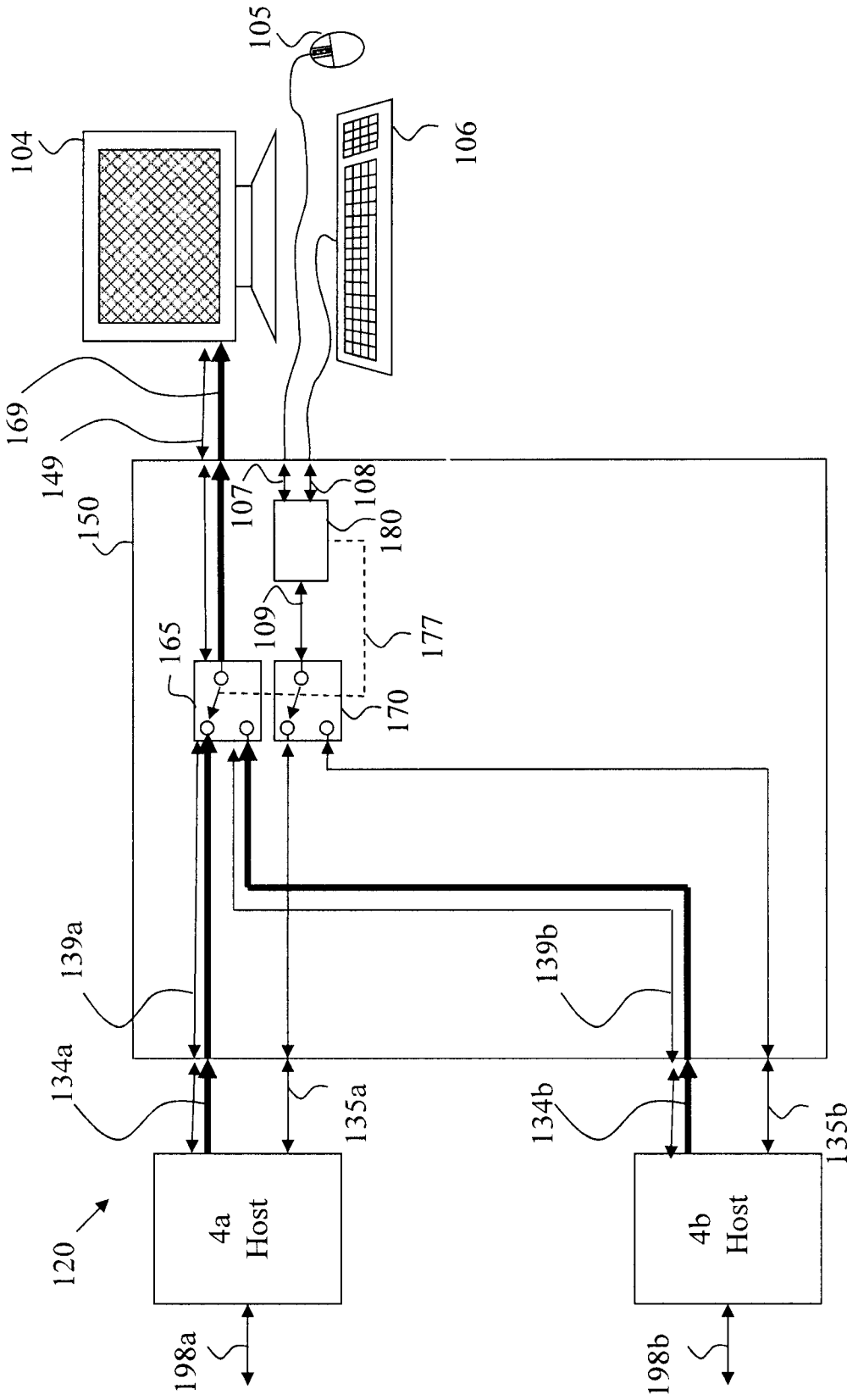


Figure 2A (Prior Art)

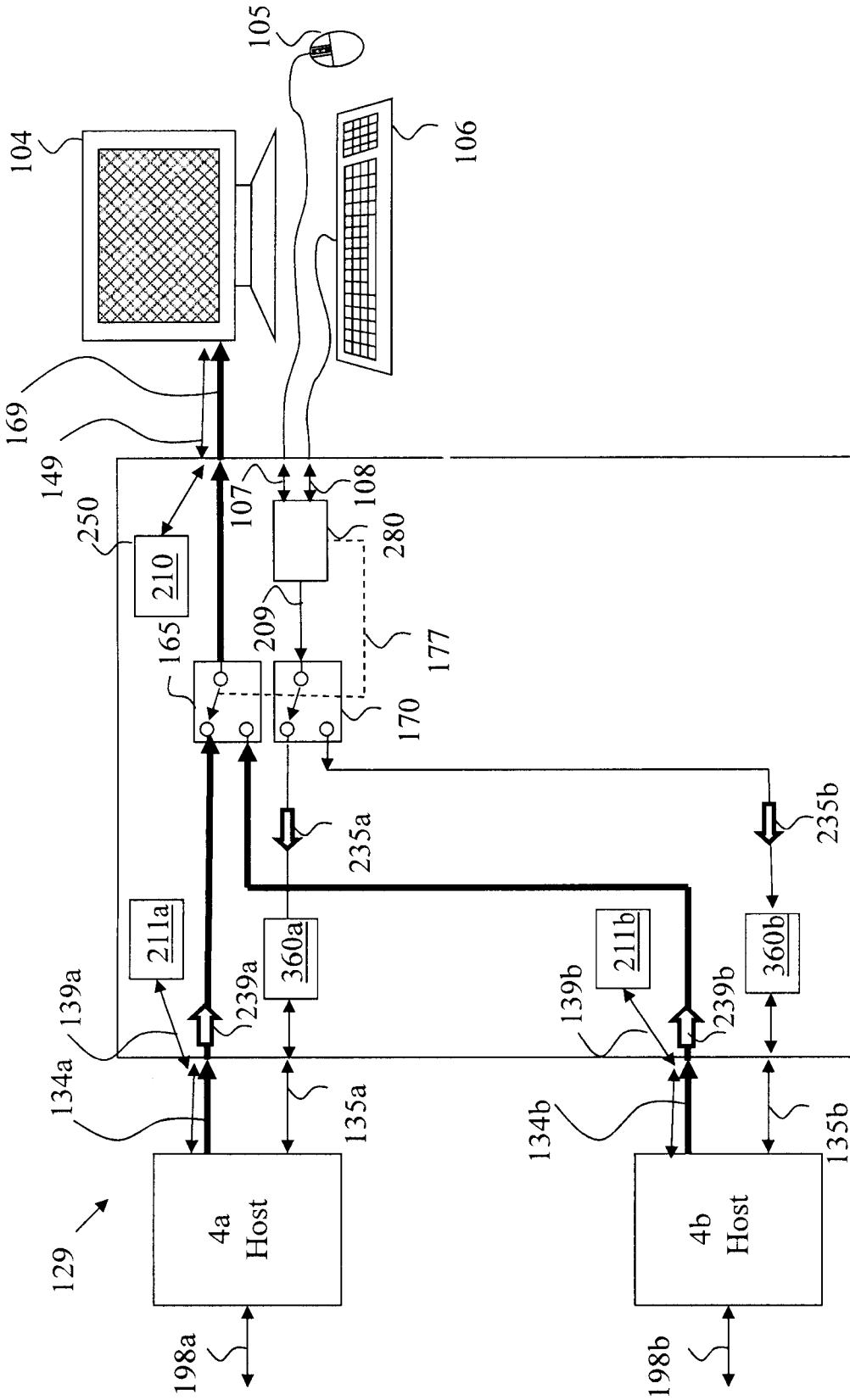


Figure 2B (Prior Art)

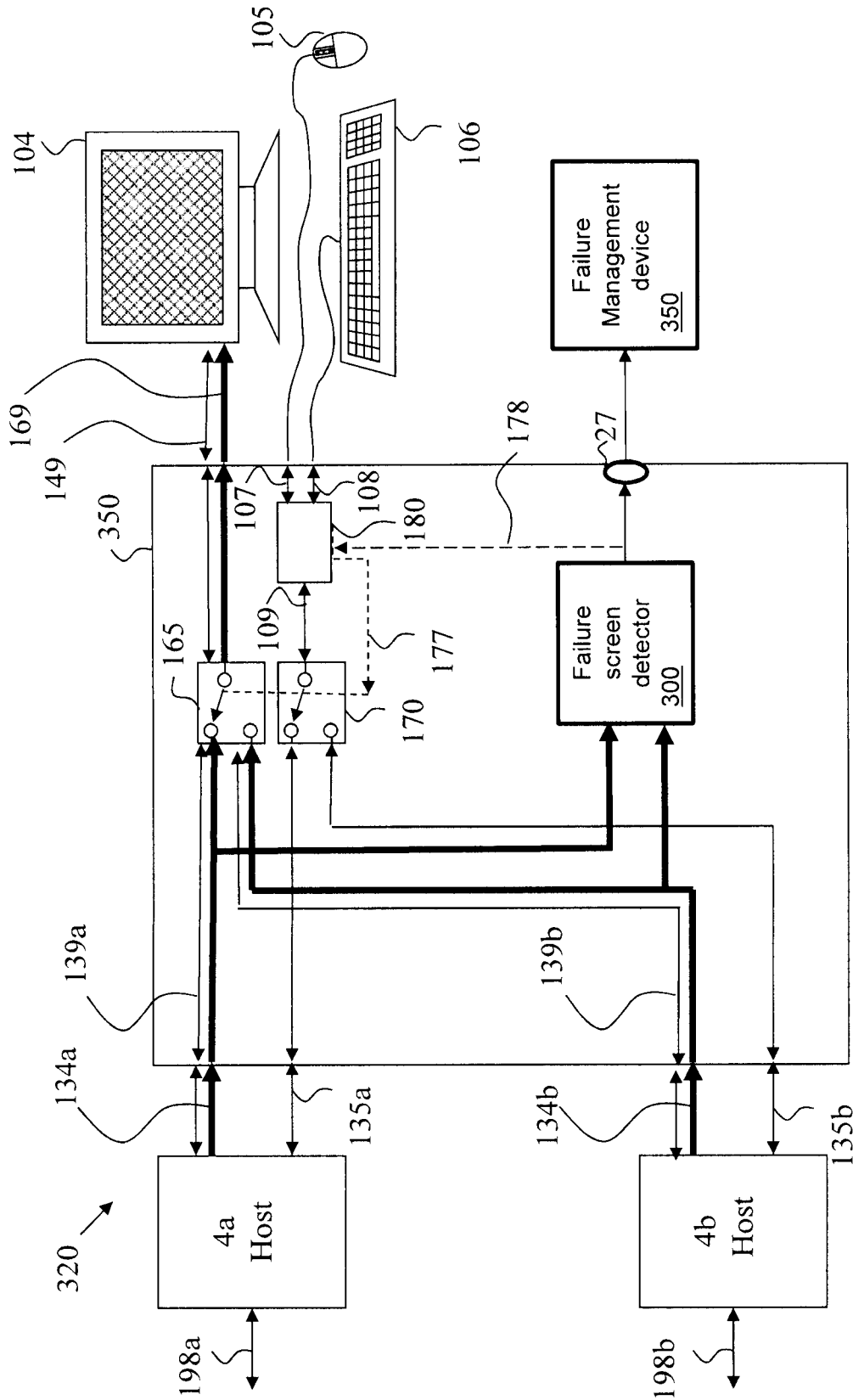


Figure 3A

6/14

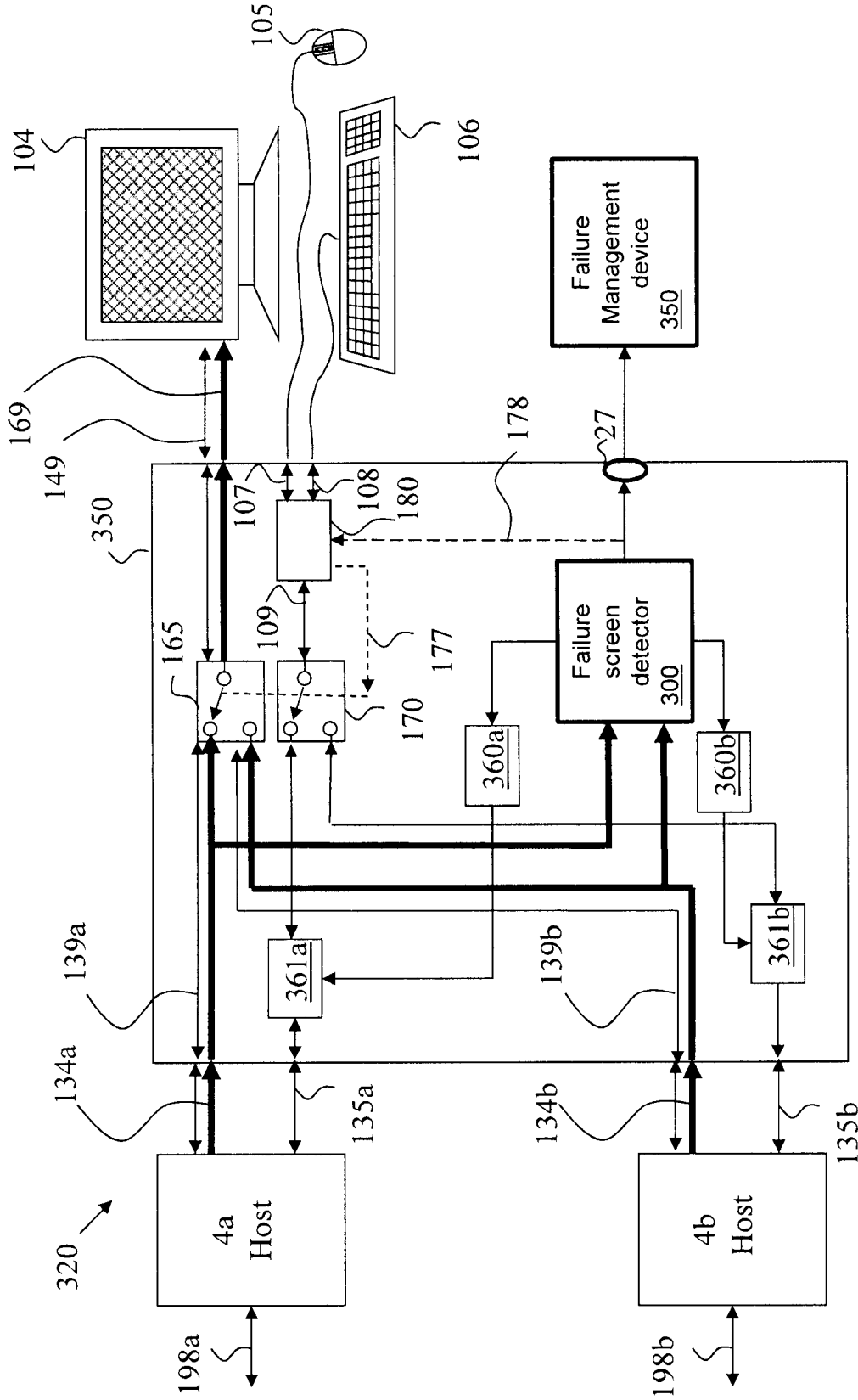


Figure 3B

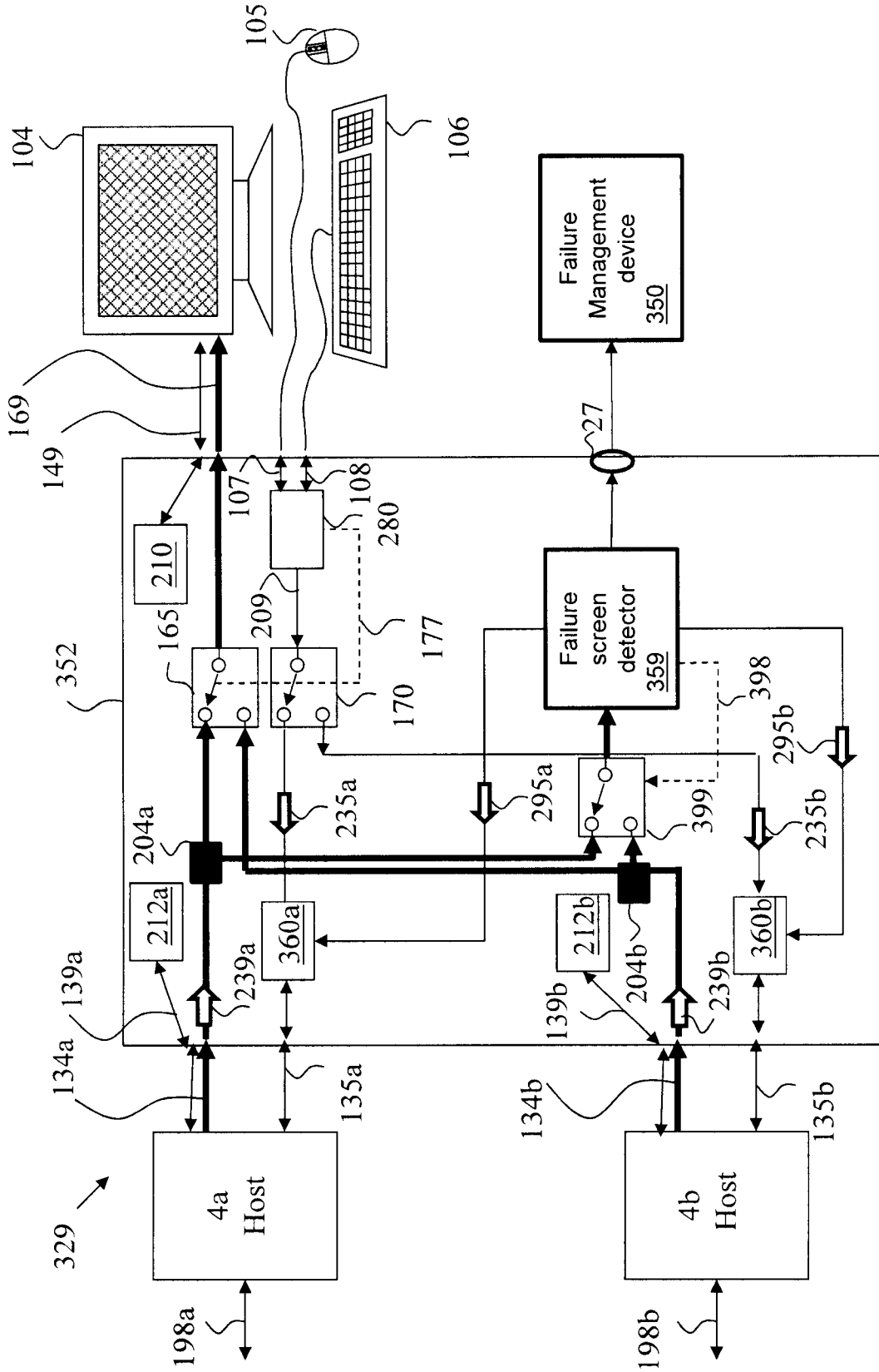


Figure 3C

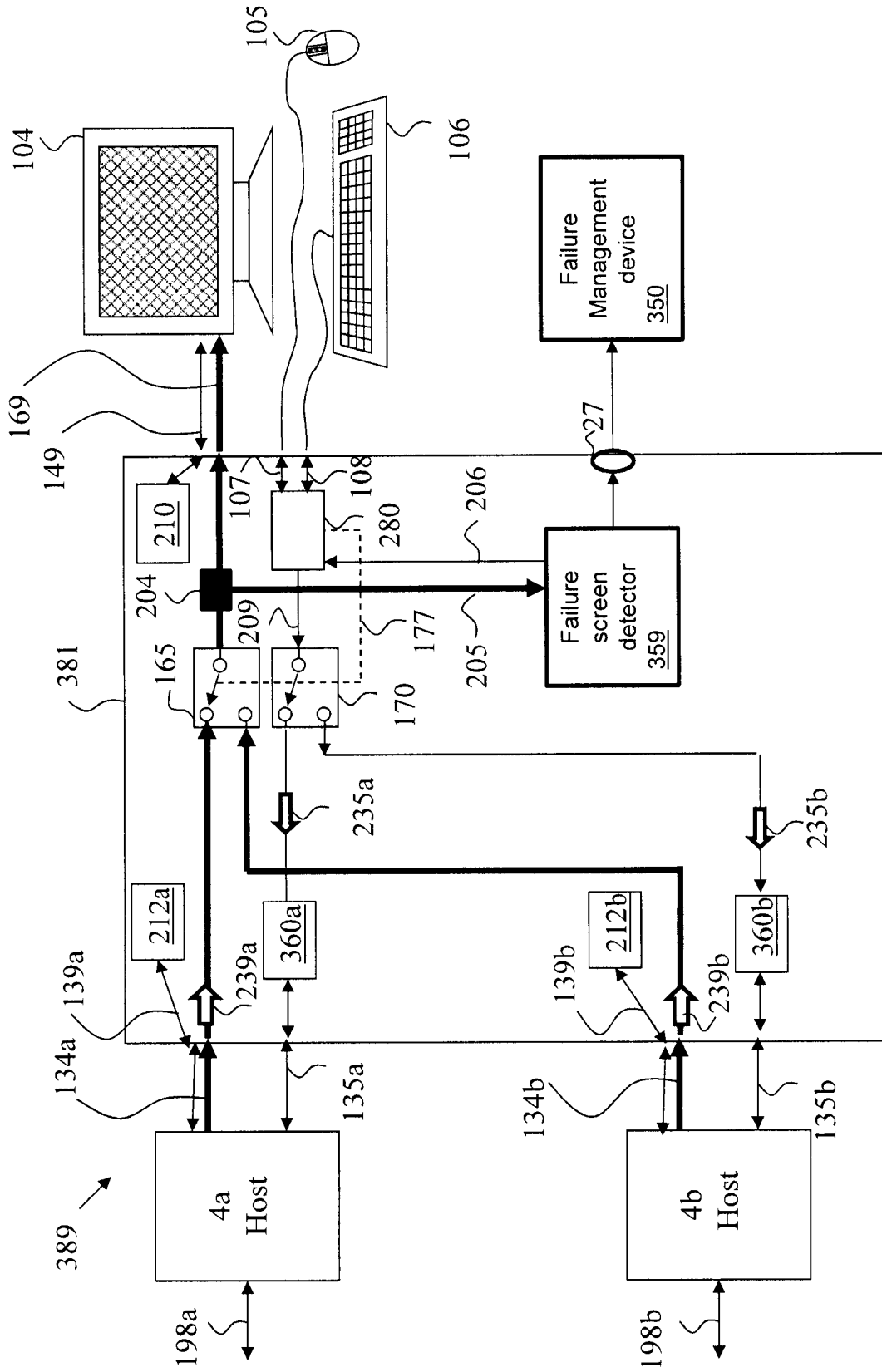


Figure 3D

9/14

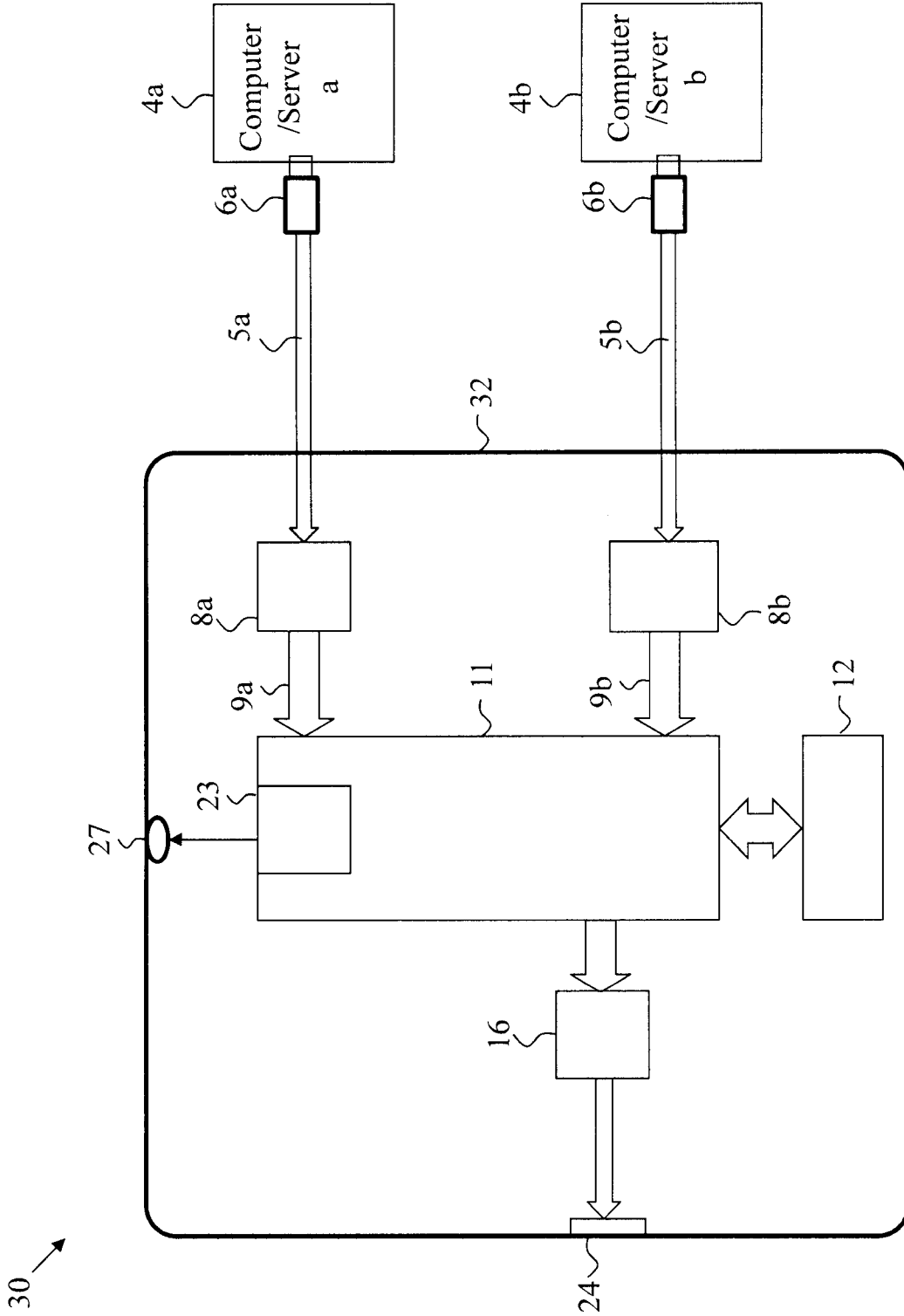


Figure 4A

11/14

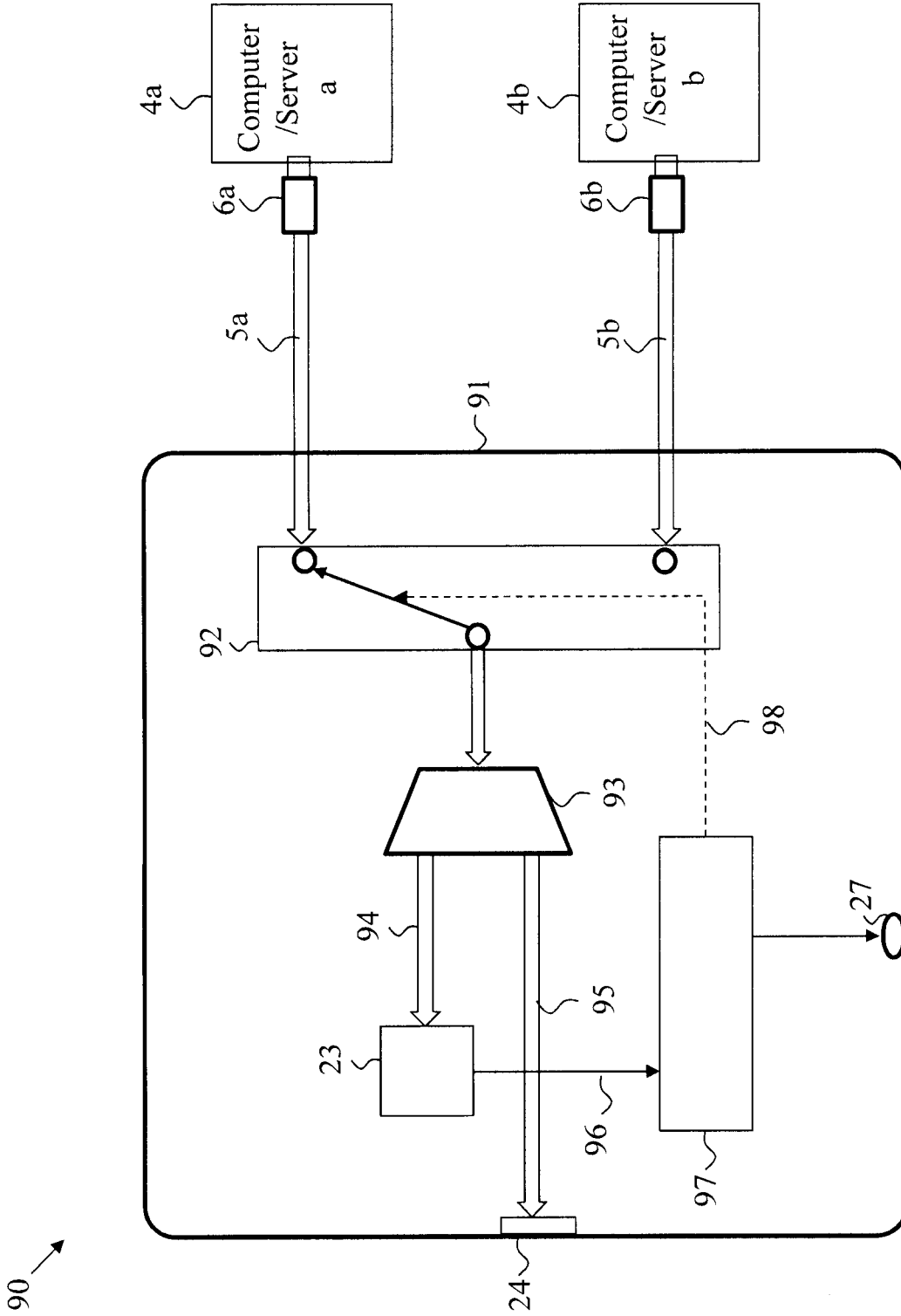


Figure 4C

12/14

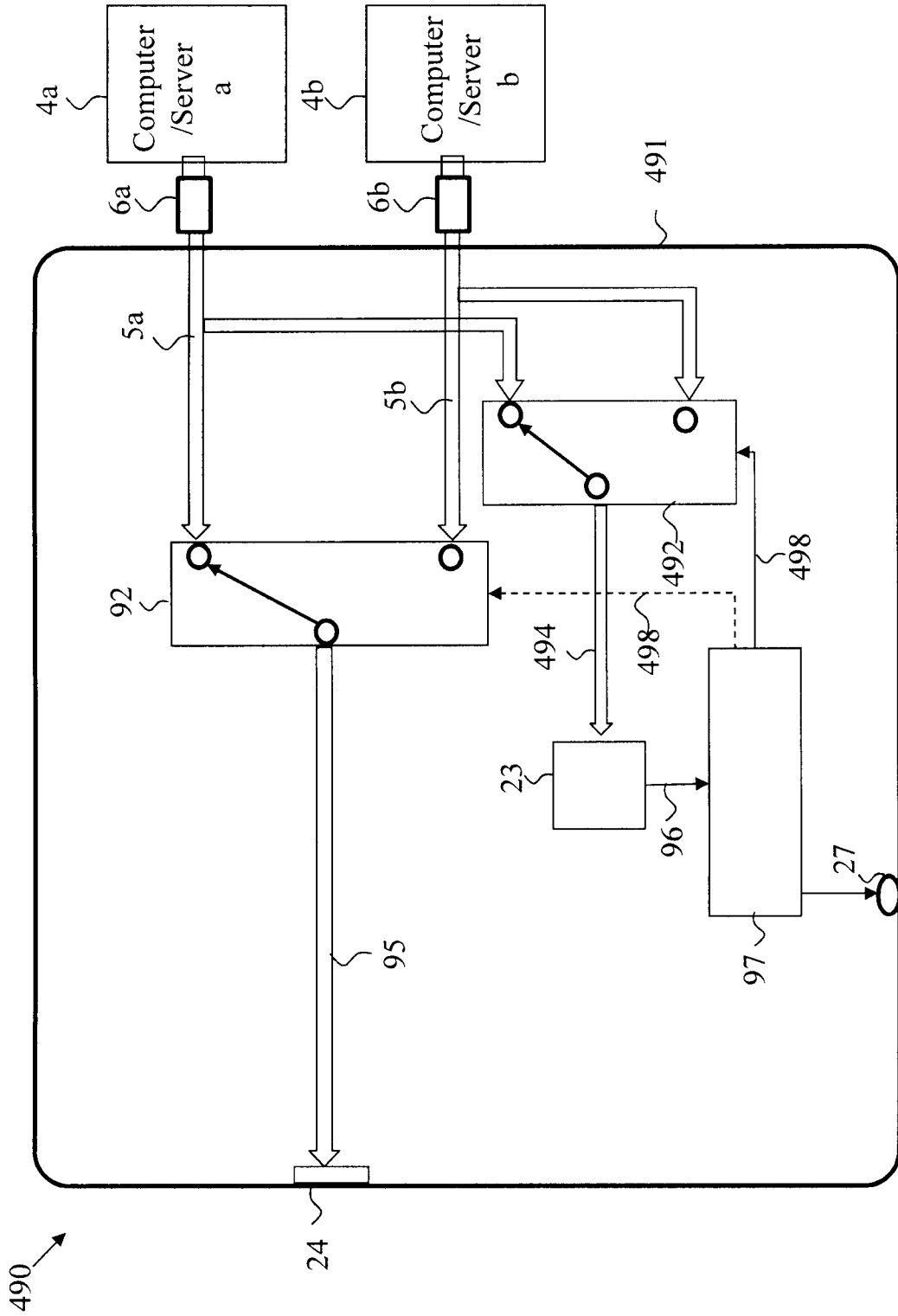


Figure 4D

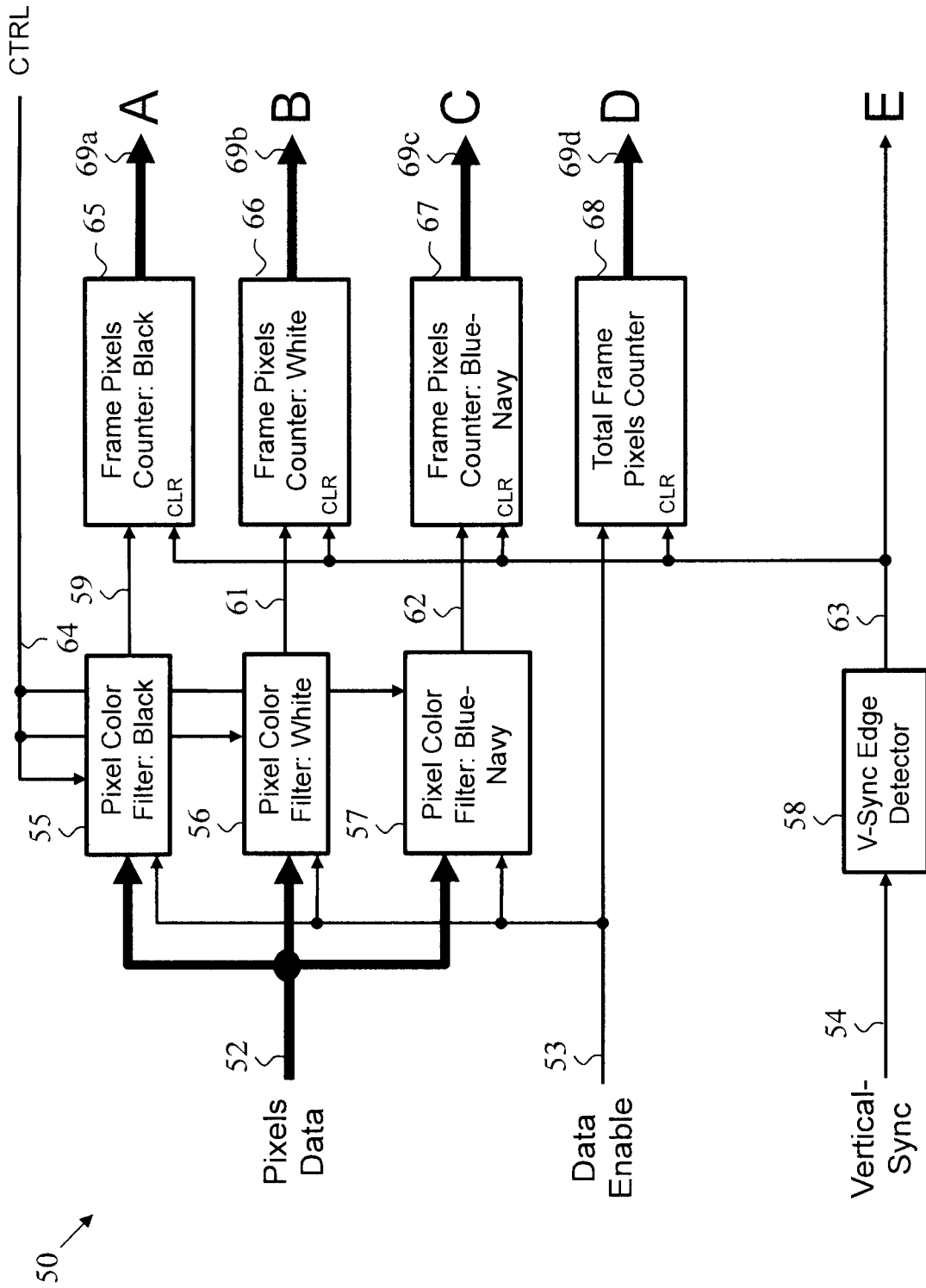


Figure 5A

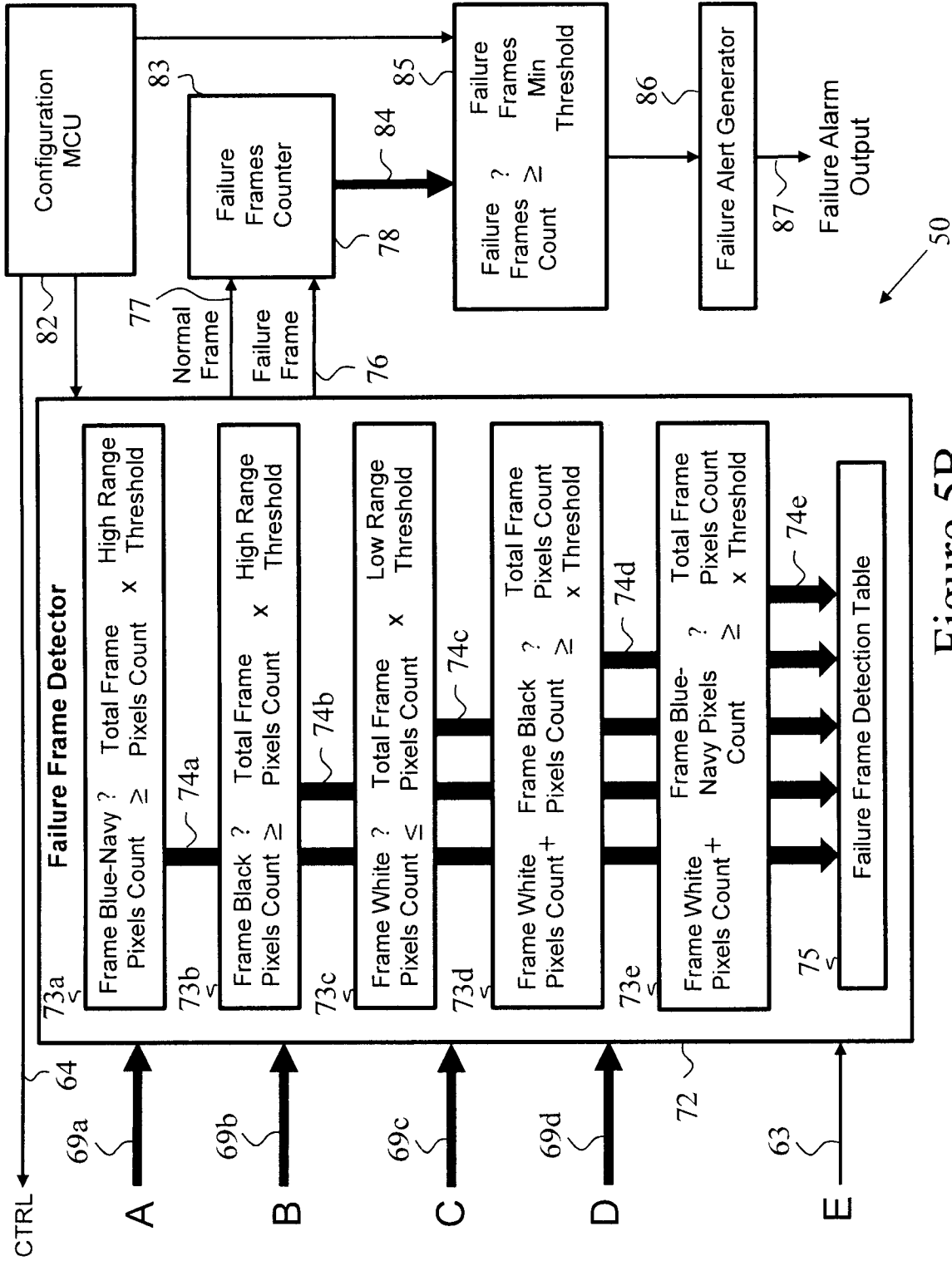


Figure 5B

