OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(12) (19) (CA) **Brevet-Patent**

(11) (21) (C) **2,087,162**

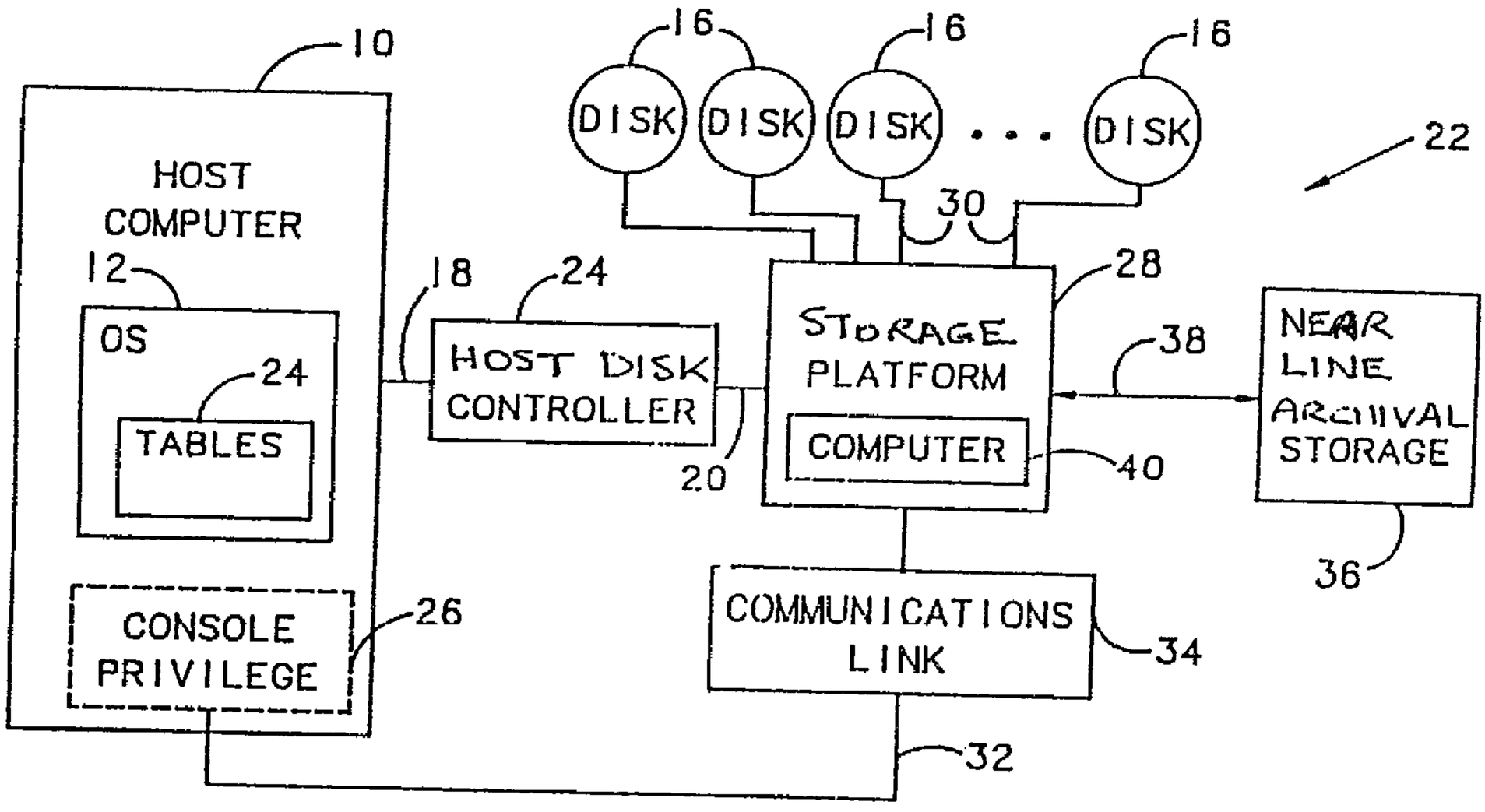
(86) 1991/07/15
(87) 1992/02/06
(45) 2001/04/24

(72) Goodlander, Theodore Jay, US
(72) Kacirek, Raul, US
(72) Sarkozy, Andras, US
(72) Hetenyi, Tamas, US
(72) Selmeczi, Janos, US
(73) STORAGE COMPUTER CORP., US

(51) Int.Cl.[5] G06F 13/16, G06F 11/00
(30) 1990/07/17 (554,127) US

(54) **SYSTEME DE STOCKAGE RAPIDE INSENSIBLE AUX DEFAILLANCES A GRANDE CAPACITE ET A CORRECTION DES ERREURS POUR ORDINATEURS BINAIRES**

(54) **HIGH-SPEED, HIGH-CAPACITY, FAULT-TOLERANT, ERROR-CORRECTING STORAGE SYSTEM FOR BINARY COMPUTERS**




(57) Cette invention prévoit la connexion d'un appareil de commande d'accès à une unité à disque entre un ordinateur central (10) et une pluralité d'unités à disque (16) pour obtenir un système de stockage à haute vitesse, de grande capacité, insensible aux défaillances, à correction d'erreurs et fonctionnant de manière asynchrone. Elle permet également d'améliorer les performances par rapport aux versions antérieures. Plusieurs canaux de commande d'unités à disque (46)

(57) This invention provides disk drive access control apparatus for connection between a host computer (10) and a plurality of disk drives (16) to provide an asynchronously operating, high-speed, high-capacity, fault-tolerant, error-correcting storage system. It also provides increases in performance over earlier versions thereof. There are a plurality of disk drive controller channels (46) connected to respective ones of the disk drives and controlling transfers of data to and from the

OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(11) (21) (C) 2,087,162

(86) 1991/07/15
(87) 1992/02/06
(45) 2001/04/24

sont connectés à des canaux respectifs des unités à disque et des transferts de commande de données vers et à partir des unités à disque, chacun des canaux de commande d'unités à disque comprenant une antémémoire/tampon (48) et une unité à microprocesseur (50). Une unité à circuit de commande et interface (42) assure l'interface avec l'ordinateur central (10) et une antémémoire centrale (54) est prévue. Un circuit logique de commande de l'antémémoire commande les transferts de données à partir de l'antémémoire/mémoire tampon (48) de la pluralité de canaux (46) de commande d'unités à disque vers l'antémémoire (54) et depuis l'antémémoire (54) vers l'antémémoire/mémoire tampon (48) de la pluralité de canaux (46) de commande d'unités à disque et à partir de l'antémémoire (54) vers l'ordinateur central (10) par l'intermédiaire de l'unité (42) de commande et interface. Une unité de traitement centrale (60) gère l'utilisation de l'antémémoire en demandant des transferts de données uniquement pour des données n'étant pas présentes dans l'antémémoire (54) et en envoyant des commandes de haut niveau aux canaux de commande (46) des unités à disque. Un premier bus (64) (de données) établit l'interconnexion entre la pluralité d'antémémoires/mémoires tampon (48) des unités à disque, l'unité (42) de commande et interface, et l'antémémoire (54) pour le transfert d'informations entre elles et un second bus (52) (d'informations et de commandes) assure l'interconnexion entre ces mêmes éléments et une unité de traitement centrale (60) pour le transfert de commandes et d'informations entre elles.

disk drives, each of the disk drive controller channels includes a cache/buffer memory (48) and a micro-processor unit (50). An interface and driver unit (42) interfaces with the host computer (10) and there is a central cache memory (54). Cache memory control logic controls transfers of data from the chache/buffer memory (48) of the plurality of disk drive controller channels (46) to the cache memory (54) and from the cache memory (54) to the cache/buffer memory (48) of the plurality of disk drive controller channels (46) and from the cache memory (54) to the host computer (10) through the interface and driver unit (42). A central processing unit (60) manages the use of the cache memory by requesting data transfers only of data not presently in the cache memory (54) and by sending high level commands to the disk drive controller channels (46). A first (data) bus (64) interconnects the plurality of disk drive cache/buffer memories (48), the interface and driver unit (42), and the cache memory (54) for the transfer of information therebetween and a second (information and commands) bus (52) interconnects the same elements with the central processing unit (60) for the transfer of control and information therebetween.

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :
G06F 13/36, 11/22, 3/153

A1

(11) International Publication Number: WO 92/01988

(43) International Publication Date: 6 February 1992 (06.02.92)

2087162

(21) International Application Number: PCT/US91/04988

(22) International Filing Date: 15 July 1991 (15.07.91)

(30) Priority data:
554,127 17 July 1990 (17.07.90) US

(71) Applicant: CAB-TEK, INC. [US/US]; 11 Riverside Street, Nashua, NH 03062 (US).

(72) Inventors: GOODLANDER, Theodore, Jay ; 4 Victoria Drive, Nashua, NH 03063 (US). KACIREK, Raul ; 11 Riverside Street, Nashua, NH 03062 (US). SARKOZY, Andras ; 26 Houston Avenue, Milton, MA 02186 (US). HETENYI, Tamas ; SELMECZI, Janos ; 1781 Wedgewood Common, Concord, MA 01742 (US).

(74) Agents: DAVIS, Anthony, G., M. et al.; Davis, Bujold & Streck, 175 Canal Street, Manchester, NH 03101 (US).

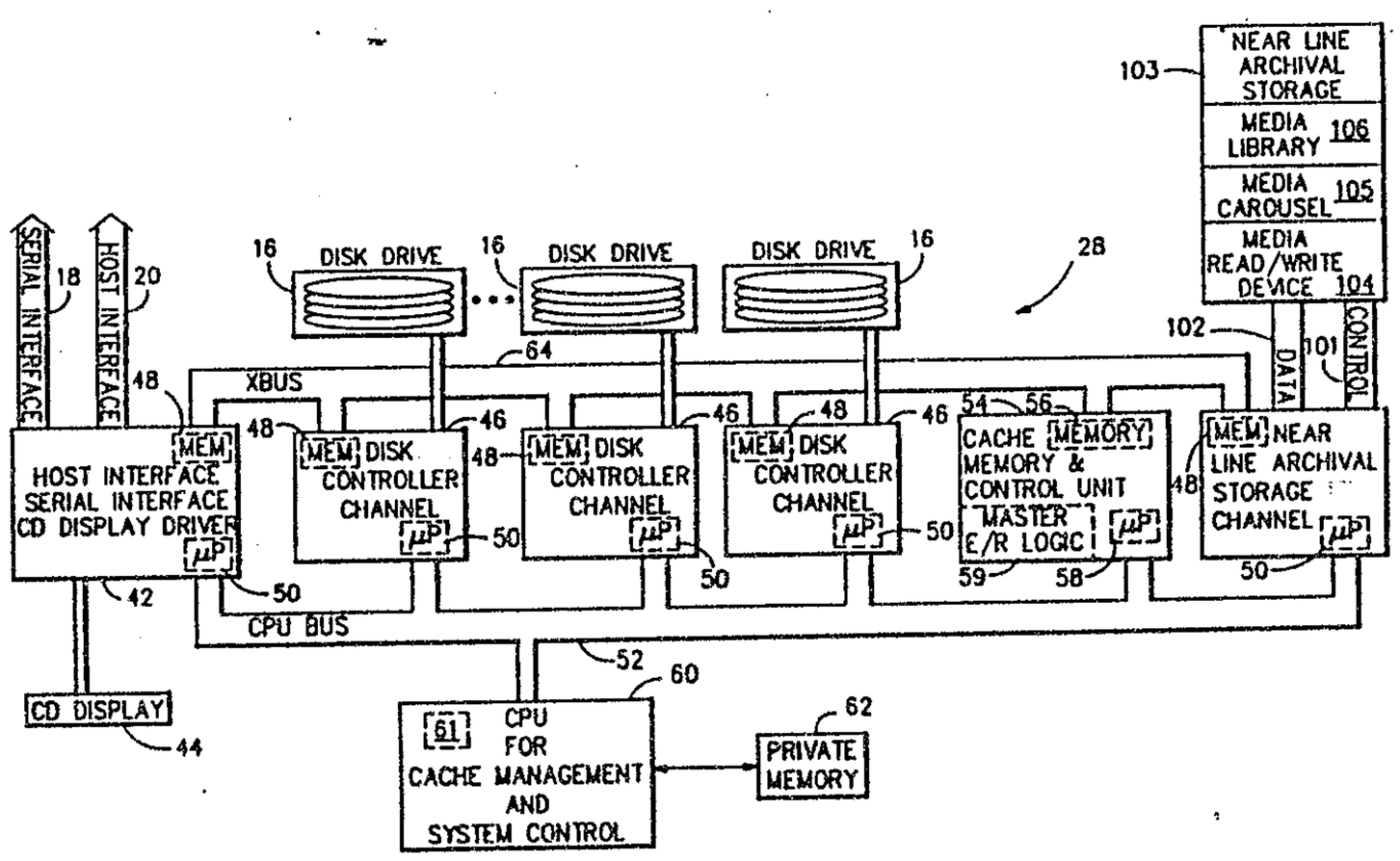
(81) Designated States: AT, AT (European patent), AU, BB, BE (European patent), BF (OAPI patent), BG, BJ (OAPI patent), BR, CA, CF (OAPI patent), CG (OAPI patent), CH, CH (European patent), CI (OAPI patent), CM (OAPI patent), CS, DE, DE (European patent), DK, DK (European patent), ES, ES (European patent), FI, FR (European patent), GA (OAPI patent), GB, GB (European patent), GN (OAPI patent), GR (European patent), HU, IT (European patent), JP, KP, KR, LK, LU, LU (European patent), MC, MG, ML (OAPI patent), MN, MR (OAPI patent), MW, NL, NL (European patent), NO, PL, RO, SD, SE, SE (European patent), SN (OAPI patent), SU, TD (OAPI patent), TG (OAPI patent).

Published
*With international search report.*

(54) Title: HIGH-SPEED, HIGH-CAPACITY, FAULT-TOLERANT, ERROR-CORRECTING STORAGE SYSTEM FOR BINARY COMPUTERS

(57) Abstract

This invention provides disk drive access control apparatus for connection between a host computer (10) and a plurality of disk drives (16) to provide an asynchronously operating, high-speed, high-capacity, fault-tolerant, error-correcting storage system. It also provides increases in performance over earlier versions thereof. There are a plurality of disk drive controller channels (46) connected to respective ones of the disk drives and controlling transfers of data to and from the disk drives, each of the disk drive controller channels includes a cache/buffer memory (48) and a microprocessor unit (50). An interface and driver unit (42) interfaces with the host computer (10) and there is a central cache memory (54). Cache memory control logic controls transfers of data from the chache/buffer memory (48) of the plurality of disk drive controller channels (46) to the cache memory (54) and from the cache memory (54) to the cache/buffer memory (48) of the plurality of disk drive controller channels (46) and from the cache memory (54) to the host computer (10) through the interface and driver unit (42). A central processing unit (60) manages the use of the cache memory by requesting data transfers only of data not presently in the cache memory (54) and by sending high level commands to the disk drive controller channels (46). A first (data) bus (64) interconnects the plurality of disk drive cache/buffer memories (48), the interface and driver unit (42), and the cache memory (54) for the transfer of information therebetween and a second (information and commands) bus (52) interconnects the same elements with the central processing unit (60) for the transfer of control and information therebetween.




+ See back of page

# HIGH-SPEED, HIGH-CAPACITY, FAULT-TOLERANT, ERROR-CORRECTING STORAGE SYSTEM FOR BINARY COMPUTERS

## Background of the Invention:

This is a continuation-in-part of application serial no. 317,048, filed 27 February 1989, issued 17 July 1990 as Patent No. 4,942,579 which is a continuation of application serial no. 057,346 filed 2 June 1989, the content of which is hereby incorporated by reference.

The application relates to memory storage systems for digital computers and, more particularly, to disk drive access control apparatus for connection between a host computer and a plurality of disk drives to provide an asynchronously operating, high-speed, high-capacity, fault-tolerant, error-correcting storage system to receive read and write requests from the host computer, read and write data from and to the plurality of disk drives, and transfer data to and from the host computer, comprising, a plurality of disk drive controller channels connected to respective ones of the plurality of disk drives and controlling transfers of data to and from an associated one of the plurality of disk drives in response to received high level commands, each of the plurality of disk drive controller channels including a cache/buffer memory and a micro-processor unit for controlling the transfers of data; an interface and driver unit interfacing with the host computer; a central cache memory; cache memory control logic controlling transfers of data from the cache/buffer memory of the plurality of disk drive controller channels to the cache memory and from the cache memory to the cache/buffer memory of the plurality of disk drive controller channels and from the cache memory to the host computer through the interface and driver unit; a central processing unit managing the use of the cache memory by requesting data transfers only with respect to ones of the plurality of disk drives where

data associated therewith is not presently in the cache memory and by sending high level commands to the plurality of disk drive controller channels to effect data transfers thereby; a first (data) bus interconnecting the plurality of disk drive controller cache/buffer memories, the interface and driver unit, and the cache memory for the transfer of information therebetween; and, a second (information and commands) bus interconnecting the plurality of disk drive controller channels, the interface and driver unit, the cache memory control logic, and the central processing unit for the transfer of control and information therebetween.

As systems employing digital computers have evolved, so have their requirements for storage systems associated therewith. Early computer systems typically had a drum or disk storage device for rapid access to required files which could not be maintained in the random-access main memory. If large quantities of data were being processed that could not fit on the drum or disk, the data were stored on removable magnetic tapes reels that could be mounted (upon request from the computer program) on a tape transport mechanism and read into the computer. Such systems were an improvement over non-computer approaches for applications requiring only occasional access to individual files such as in banking, bookkeeping, payroll, insurance, and similar undertakings. As many applications became more computationally intensive and systems became more complex, the limitations and requirements of the storage systems needed to change. In most instances, however, storage technology has not kept up with the needs of the computer systems. In the original digital computers, a program and its data was loaded, executed, and printed out the answers. The programs involved were straight forward and relatively easy to program. If a program contained an error (i.e. a "bug"), it simply failed to produce an executable program when compiled or assembled and failed to produce an answer when ultimately

executed. In a computer system where multiple computers are linked together and run multiple programs on multiple priority levels under the control of an interrupt structure, the problem is not so easy. A program improperly programmed in so-called "re-entrant" coding may run for days or even weeks before the right set of circumstances cause an error. Not only that, such systems are designed to run continuously and only produce results in response to some external stimulus. A bug in such a system may actually be fatal to the system and cause the entire computer to stop executing its programs. If the system in question is an air traffic control system, for example, the term "fatal" bug could be true both literally and figuratively.

If there is a problem with the disk drive system, it should be transparent to the users and the disk drive system should continue to operate despite errors. Moreover, it should correct the errors so as to provide reconstructed, error-free data. Likewise, if the disk drive system is unable to contain all the data files, the required data should be retrieved automatically from a near line archival storage system in as short a time as possible. Preferably, the various functions should be under the control of an artificial intelligence type of system so that needs of the users can be learned and anticipated so as to optimize the system's performance capabilities.

A given system's overall performance capability is the result of the integration of a number of independent technologies whose individual growth in performance/capability over time differs markedly from each other. Figure 1(a) to (g) shows the historical performance of computer systems' underlying technologies. Figure 1a shows the expodential growth in semiconductor component performance. The factors behind this are well known and include advances in process technology that increase circuit density and speed. Shrinking geometries

and increased wafer yields combined with circuit design innovations mean semiconductor performance should continue its expodential growth. Figure 1b reflects the expodential growth in CPU hardware performance as measured in MIPS. CPU's are the direct beneficiaries of semiconductors as well as circuit design improvements and architectural innovations such as massively parallel CPU's. Figure 1c shows the trend in performance for operating systems. OS performance is being flattened by several factors such as the additions of user interfaces, graphics support, and the sheer growth in size over the years, which has made OS's one of the most voracious consumers of computer resources. Figure 1d shows the capacity/performance improvements of disk drives. This curve could be best described as "leap-linear." Disk drive device performance and capacities tend to grow linearly until a new technological event occurs. Such events in the past have been the introduction of sealed disk Winchester technology in the early 70s, the introduction of thin film heads and plated media in the 80s, and, in the 90s, the general introduction of 5400 and 7200 RPM drives to cut latency delays and contact recording technology that may push track densities to 100,000 per inch. Figure 1e demonstrates the performance of I/O systems. This curve represents the composite effect of CPU and controller hardware, operating systems, and disk drives. It should be noted that the expodential growth in semiconductor performance has not been reflected in I/O system performance, which has seen near linear growth. Figure 1f reflects a commonly understood phenomena with applications programs that, over time, additions, changes, and maintenance to the program tend to lead to a decrease in its performance. Figure 1g demonstrates that, over all systems performance has showed continued improvement, but at a much slower rate than its underlying technologies. In fact, without hardware upgrades and improvements over all system performance

declines in response to the performance of operating systems and application programs.

Figure 1(h) shows a typical prior art computer system employing disk drives for storage. The host computer 10 (i.e. the one interfacing with the computer operators) includes an operating system 12. As known to those skilled in the art, the operating system is a set of computer programs that run continuously while the computer has its power on. The operating system controls all the functions of the computer including requests for operating portions of the memory, error response, and input/output (I/O) requests. The computer 10 has a disk controller 14 connected thereto and the disk controller 14, in turn, is connected to four disk drives 16. In use, an applications program (not shown) makes a request for data from the operating system 12. The location of the data is completely transparent to the applications program; that is, the applications program has no idea where the data is physically located. At system setup time (or possibly subsequently through operator input), the locations of the data is stored in tables (not shown) which are part of or accessible by the operating system 12. Knowing from the tables that the requested data is on a particular disk drive 16 at a particular track between starting and ending sectors, the operating system 12 outputs a disk read request on line 18 to the disk controller 14. The disk controller 14, in turn, then issues a read request to the appropriate disk drive 16 on its connecting line 20 which causes the read head (not shown) within the disk drive 16 to move to the designated track and then read data and output it to the disk controller 14 on the line 20 from the starting sector to the ending sector. When the data has been received by the disk controller 14 (into an appropriatecache/buffer memory, the operating system 12 is informed by an appropriate signal on line 18.

As can be appreciated, if one wants the operating system 12 to do more, the programming of the operating

system 12 must get more complex. Given the present state of complexity of the typical operating system and the capabilities of the average systems' computer programmer with respect to such esoteric matters as re-entrant coding and "run anywhere" coding, to ask the operating system to do more is to ask for trouble because of information handling bottle necks.

There is also the problem of system overhead. If you ask the operating system to do more, it will add to the overhead of the operating system and, therefore, to the overhead of every program which accesses it.

For any given OS and computer system, implementation of any real time function will cause the OS to consume a large portion of the computing resource, rapidly degrade the performance of the system from the user's perspective, and severely limit the work product computing potential.

As those skilled in the art will also readily recognize and appreciate, even if the penalty of added overhead is made to the operating system so as to achieve the convenience in other areas, such an approach includes no means of ever reducing the added overhead.

File Maintenance, Management, and Archival Copy (FMMAC) are tasks essential to the reliability, useability, and integrity of computer stored data. These tasks are now performed by Operating Systems functions, separately run applications programs, operator or system manager manual intervention, or a combination of these techniques.

These FMMAC tasks almost always require a manual operator decision to initiate and complete. Often they require the computer system to be taken offline and therefore not available to users during the time it takes to complete these tasks. Some larger, sophisticated Operating Systems allow a designated File to be taken offline leaving most of the computer resource available. However, manual intervention is still required to initiate file maintenance and archival copy.

Because these crucial FMMAC tasks rely on manual intervention, arbitrary circumstances and schedules, the predictability of these tasks being performed is low. This is especially true outside of centralized "Mainframe" Computer centers (FMMAC tasks are typically performed here by a dedicated maintenance shift at great additional operating expense). However, most computers (and by extension, most computer stored data) are not located inside "computer centers" or have the benefit of dedicated file maintenance staffs. Therefore the reliability, usability, and integrity of most computer stored data now rests on human nature and motivation and the dubious assumption that the circumstances surrounding the computer system itself are immune from intervening events such as device failures or rush jobs that take priority over FMMAC tasks.

Continuous duty computer systems such as Real Time monitoring and control systems or Online Transaction Processing systems present additional barriers to FMMAC tasks. In theory, no time is available to perform the tasks. In reality, such systems or files are simply shut off for FMMAC. In "Critical Mission" applications (for example Nuclear Power Plant Monitoring and Control) the FMMAC problem is often abated by duplicating hardware systems at great economic cost.

In the above-referenced parent application of which this is a continuation-in-part, a high-speed, high-capacity, fault-tolerant, error-correcting storage system was disclosed which provides a solution for many of the above-described needs of modern computer systems (both military and commercial). Since its original filing which contained the best mode as contemplated at that time, continued work has developed new embodiments which include novel improvements over the teachings contained therein. In particular, the performance and capacity potential has been significantly increased as has its flexibility as a

system component. Moreover, additional improvements to the overall storage system have been incorporated therein.

Wherefore, it is an object of this application to provide significant and patentably distinct improvements to the high-speed, high-capacity, fault-tolerant, error-correcting storage system of the above-referenced parent application of which this is a continuation-in-part.

It is another object of this application to provide significant and patentably distinct improvements to high-speed, high-capacity, storage systems for digital computers in general.

It is yet another object of this present invention to provide a means to maintain, manage, and archive copy files on an automatic and User transparent basis and to provide high predictability for the reliability, usability, and integrity of computer stored data.

Computer stored Files are generally classified by their application use such as "payroll files" or "inventory files" much in the same way that a typical office filing cabinet and the folders inside would be labeled. Those skilled in the art will recognize that files can also be classified by their usage in time. That is to say that some files are used more often than others and can be reliably classified on that basis:

| <u>CLASSIFICATION</u> | <u>USAGE IN TIME</u> |
|---|---|
| Continuous: | A file that is heavily accessed. An example would be a directory or VTOC file. Accesses could be seconds, minutes, or even hours apart. |
| Systematic: | A file opened & accessed as a consequence of other files being used. |
| Periodic: | A file that is used on a predictable schedule such as daily, weekly, monthly, 1st of month, 15th of month, etc. |
| Occasional: | A file whose usage in time cannot be predicted or classified by the above criteria. |

Transient: A file created and deleted in a short time interval. A scratch file not intended for any future use.

Summary:

The foregoing objects have been achieved in a computer system including a host computer connected to access a plurality of data storage units and including an operating system having tables of file locations of data on the data storage units, by the improvement of the present invention comprising, data storage unit access control means connected between the host computer and the plurality of data storage units for receiving read and write requests from the host computer, for reading and writing data from and to the plurality of data storage units, and for transferring data to and from the host computer; communications link means connected between the data storage unit access control means and a console privilege interface of the host computer operating system for obtaining and providing the data storage unit access control means with data file location information contained within the tables; and, computational logic means contained in the data storage unit access control means for using the communications link means to obtain file location information about a data request from the host computer whereby the operating system continues to operate with respect to providing transparency about data locations to user programs operating within the host computer.

In the preferred embodiment, the computational logic means includes learning logic so that once the location of a requested data on the plurality of data storage units has been obtained using the communications link means, it is known and remembered by the computational logic means.

In a computer system wherein the computer system includes near line archival storage apparatus of an automated variety wherein off-line storage media are loaded for reading and writing on request by robotic

mechanisms so as to avoid the necessity of operator intervention, the preferred embodiment includes the computational logic means including logic for using the communications link means to obtain location information about files on the near line archival storage apparatus. As a further feature of the preferred embodiment, the computational logic means includes learning logic so that once the location of a requested file on the near line storage apparatus has been obtained using the communications link means, it is known and remembered by the computational logic means. Also, the learning logic of the computational logic means is of an artificial intelligence variety which learns through experience so that if a particular file from the near line archival storage apparatus is repeatedly used at a given time and day of the month the learning logic learns that from experience and begin loading the file from the near line archival storage apparatus in anticipation of an expected request so that when the request is made, the file is already read in and available for use. Such files are handled in such a manner by computational logic to appear to the user CPU as a virtual array.

The near line archival storage apparatus is managed in such a way by the computational logic means that it virtually extends the maximum capacity of the physical array drives to include the capacity of the entire archive thus creating the appearance of a disc drive of boundless capacity without degrading performance.

In addition this solves the aforesaid backup problems related to file maintenance, management and archival copy.

Preferably, the computational logic means also includes testing logic for testing the plurality of data storage units in a background mode on a time-available basis.

The preferred data storage unit access control means comprises, a plurality of data storage unit controller channel means connected to respective ones of the

plurality of data storage units for controlling transfers of data to and from an associated one of the plurality of data storage units in response to received high level commands, each of the plurality of data storage unit controller channel means including a cache/buffer memory and a micro-processor unit for controlling the transfers of data and status; interface and driver unit means for interfacing with the communications link means and the host computer; a central cache memory; cache memory control means for controlling transfers of data from the cache/buffer memory of the plurality of data storage unit controller channel means to the cache memory and from the cache memory to the cache/buffer memory of the plurality of data storage unit controller channel means and from the cache memory to the host computer through the interface and driver unit means; central processing unit means for managing the use of the cache memory by requesting data transfers only with respect to ones of the plurality of data storage units where data associated therewith is not presently in the cache memory and for sending high level commands to the plurality of data storage unit controller channel means to effect data transfers thereby; first (data) bus means interconnecting the plurality of data storage unit controller cache/buffer memories, the interface and driver unit cache/buffer memory, and the cache memory for the transfer of information therebetween; and, second (information and commands) bus means interconnecting the plurality of data storage unit controller channel means, the interface and driver unit means, the cache memory control means, and the central processing unit means for the transfer of control and information therebetween. The preferred data storage unit access control means includes, an alphanumeric display; and, driver means for driving the display.

The preferred interface and driver unit means comprises, first bus driver means for connecting to and interfacing with the first bus; host computer interface

driver means for connecting to and interfacing with the host computer; communications link interface means for connecting to and interfacing with the communications link means; a first unidirectional FIFO connected between the first bus driver means and the host computer interface driver means for receiving inputs from the first bus intended for the host computer; a second unidirectional FIFO connected between the host computer interface driver means and the first bus driver means for receiving inputs from the host computer intended for the first bus; local micro-processor means for controlling the first unidirectional FIFO, the second unidirectional FIFO, and the host computer interface driver means; and, connector means for connecting the local micro-processor means to the second bus.

In the preferred embodiment, each of the plurality of data storage unit controller channel means comprises, first bus driver means for connecting to and interfacing with the first bus; data storage unit interface driver means for connecting to and interfacing with an associated one of the plurality of data storage units; a cache data buffer; data semaphore means for preventing simultaneous access to the data buffer by the first bus driver means and the data storage unit interface driver means; buffer address register means for providing a current active address into the data buffer; local micro-processor means for controlling the data semaphore means and the data storage unit interface driver means; and, connector means for connecting the local micro-processor means to the second bus.

Figure 1(i) demonstrates the performance potential of the present invention as well as the increasing gap between CPU and storage I/O performance with prior art.

Description of the Drawings:

Figure 1(a) to (i) show the historical performance of computer systems1a-g, is a simplified block diagram of a

prior art approach to a disk-based memory system 1(h), and the performance potential of the present invention 1(i).

Figure 2 is a simplified block diagram of a platform-based disk memory system according to the present invention.

Figure 3 is a simplified block diagram of the disk memory platform of the system of Figure 2.

Figure 4 is a simplified block diagram of the host interface block of the disk memory platform of Figure 3.

Figure 5 is a simplified block diagram of the one of the disk controller channels of the disk memory platform of Figure 3.

Figure 6 is a simplified drawing employed for an example to show the dual asynchronous operation of the present invention.

Description of the Preferred Embodiment:

An improved overall computer system according to the present invention employing both disk and near line archival storage is shown in Figure 2 where it is generally indicated as 22. The system 22 has a host computer 10 containing an operating system 12 with its tables 24. There is also a console privilege interface 26 by means of which outside user consoles (not shown) can be used to access the host computer operating system 12. There is once again a disk controller 24 since there is no change to the operating system 12 and the operating system 12 is set up to interface with the disk controller 24. Rather than being connected directly to the disk drives 16 as in the prior art approach of Figure 1(i), however, the single line 20 from the disk controller 24 is connected to an intelligent disk platform 28. The disk platform 28 is then connected to interface with the disk drives 16 through lines 30. Additionally, the disk platform 28 has a bi-directional connection 32 through a communications link 34 to the console privilege interface 26. In the preferred embodiment as applicable for large-scale storage

2087162



systems, there is also near line archival storage apparatus 36 connected to the disk platform 28 through line 38. To perform within the system 22 of this invention, the near line storage apparatus 36 should be of an automated variety selected from any of a number well known in the art where off-line storage media are loaded for reading and writing on request by automated mechanisms so as to avoid the necessity of operator intervention.

To accomplish its unique improvements over the prior art, the disk platform 28 includes its own computing capability as represented by the computer block 40. As will be seen shortly, the computer 40 may, in fact, comprise multiple processing units; but, for the present it is sufficient to note that the disk platform 28 is not the "dumb" controller 14 of the prior art. Employing the bi-directional connection 32 through the communications link 34 to the console privilege interface 26, the computer 40 can find the location of data from the tables 24 of the operating system 12. The location of data within the disk drives 16 or the near line archival storage apparatus 36 is, therefore, transparent to both the operating system 12 and the applications programs requesting it. If requested data is located on the near line archival storage apparatus 36, it can be retrieved automatically and then be relayed to the operating system 12 just as if it was on one of the disk drives 16. More importantly, the preferred computer 40 is of the self learning variety which learns through experience. Thus, for example, if a particular file from the near line archival storage apparatus 36 is used at a given time and/or day each month (as in preparing a payroll), the logic of the computer 40 will learn that from experience and begin loading the file from the near line storage apparatus 36 in anticipation of the expected request so that when the request is made, the file is already read in and available for use. Thus, the overall system performance of the system 22 is not only improved over the

prior art for a given level of file location transparency to the users; but, additionally, the overhead drops as the system learns the repeated patterns of use associated with its users. In addition, whereas the prior art approach of Figure 1(i) could only do system and equipment diagnostics by taking the computer 10 off-line or by increasing the complexity and overhead of the operating system 12 once again by having the operating system 12 perform such functions in a background mode, the storage system and equipment diagnostics are now performed by the computer 40 located within the disk platform 28 on a continuing and time-available basis. As will be seen from the description which follows, when the disk drives 16 have a fault or error, any errors and problems found can be corrected or at least pin pointed for operator correction and replacement without taking the system 22 off line or causing any significant degrading of the performance thereof.

Having thus described a preferred overall system approach to storage systems according to the present invention, attention will now be turned to a preferred construction of the disk platform 28. As reference to the above-referenced parent application of this application will reveal, the essence of that invention was the asynchronous accessing of the various disk drives 16. That system employed individual disk channel controllers for respective ones of the disk drives 16 and each of the disk channel controllers included a cache/buffer memory. All the disk channel controllers were controlled by control signals from a common control computer and both data and the control signals employed the same bus. As depicted in Figure 3, the preferred disk platform 28 of this disclosure has been restructured to greatly improve the performance while, at the same time, also substantially reducing the complexity whereby the reliability is increased and the implementation thereof is greatly simplified with attendant known benefits.

2087162



In Figure 3, line 18 is labelled as the "SERIAL INTERFACE" and line 20 is labelled as the "HOST INTERFACE". In a tested embodiment, the SERIAL INTERFACE of line 18 is an RS-232 interface and the HOST INTERFACE of line 20 is a SCSI (small computer system interface) interface. This choice was as a result of the availability of commonly-used equipment for use in testing only and those skilled in the art will readily recognize and appreciate that the same techniques being described here by way of example could be accomplished employing other hardware interface methods and apparatus known in the art, or yet to be developed. In this regard, the improvements of this invention are both universal and adaptable. The two lines are connected to a interface and driver unit 42 which provides the host interface, serial interface, and LCD display driver functions as indicated therein. The logic and apparatus of interface and driver unit 42 is also connected to a display 44. The display 44 is not an absolute requirement; but, is preferred so that messages can be provided to an operator as, for example, in the event of a detected malfunctioning disk drive 16 which has been removed from operation and should be replaced in order to restore full system capability.

There are a plurality of disk controller channels 46 connected to respective ones of a plurality of disk drives 16. At this point, it is reasonably anticipated that up to twenty-three disk drives 16 can be supported by the disk platform 28. It is preferred for error detection and correction optimization that at least one extra disk controller channel 46 and associated disk drive 16 be up and running at all times to be used as a "hot" replacement in the event of a detected error or potential problem. While it will be remembered from the parent case that apparatus according to the general approach of which this is an improvement can continue to operate with detected errors by correcting them on the fly, such dynamic correcting will degrade the performance of the system.

Thus, it is preferable to dynamically substitute a disk channel and let the problem channel be fixed by service personnel off-line. When repaired, that channel can then become the hot replacement. Each of the disk controller channels 46 again has a cache/buffer memory 48 associated therewith. Where the memory of the disk controller channels of the parent case performed the dual function of a memory buffer into which to read and from which to write with respect to the disk drive connected thereto and a cache memory, the memories 48 of this invention in general perform the buffer function. It is preferred that the memories 48 be sized to be involved in more than one transfer at a time so that at least one stream of data can be read into the memory 48 while another stream of data is being written out of the memory 48. Another important distinction between the approach of this invention and that of the parent application is the inclusion of a micro-processor 50. As will be remembered, the disk controller channels of the parent case were dumb; that is, they did not include any computing power. Thus, the disk controller channels of the parent case had to be fed a stream of low level commands, resulting in a large amount of non-data traffic sharing the bus with the data. That, of course, reduced the overall throughput of data along the bus. In the "intelligent" disk controller channels 46 of this approach, the disk controller channels 46 are provided only with high level command signals and accomplish the balance of the indicated read and write functions internally with the micro-processors 50. Thus, the amount of non-data traffic on the CPU bus 52 is greatly reduced, allowing more available time for the transfer of data -- which, of course, should be the primary function thereof for optimum system performance.

A significant element of this application is the cache memory & control unit 54. While the memories 48 of the disk controller channels 46 in the tested embodiment are each 8K bytes in size, the cache memory 56 of the

cache memory & control unit 54 is 16M bytes. The cache memory & control unit 54 also includes its own microprocessor 58 and master error recovery logic 59.

Finally, the preferred disk platform 28 includes a central processing unit (CPU) 60 with its own private memory 62. The CPU 60 is employed for cache management and system control functions in a manner to be described shortly. As shown in Figure 3, the above-described components with the exception of the CPU 60 are interconnected to communicate with one another by the XBUS 64 and interconnected to communicate with the CPU 60 by means of the CPU bus 52.

As mentioned above, the main point of novelty of the parent case was that data was transferred to and from the various disk drives asynchronously. In this invention, there are two layers of asynchronous operation -- which greatly enhances the overall system performance. Not only is the transfer of data between the memories 48 and the disk drives 16 done asynchronously as in the prior disclosure; but, additionally, transfers into and out of the cache memory 56 is done asynchronously. Because of its size (i.e. 16M bytes) the cache memory 56 can be employed for the transfer of multiple requests from the host computer 10 at one time. Because of access times of the disk drives 16, lengths of transfers requested, and/or data portions already in the cache memory 56, the requests to read and write data from the CPU 60 to the disk platform 28 may not be responded to in the order received. As soon as a request is completed, no matter its position in the request sequence, the CPU 60 is informed that it is completed. Thus, throughput as seen by the host computer 10 is greatly improved.

The CPU 60 includes an embedded array disk operating system 61 and employs its private memory 62 to keep track of the contents of the cache memory 56 so that it can respond optimally to requests from the host computer 10. As stated earlier, the CPU 60 in this system only issues

high level commands to the disk controller channels 46 so that the multiple, low-level command approach of the parent case, which occupied valuable time on the CPU bus 66, is eliminated. The micro-processors 50 each contain firmware that not only performs the actual low-level command steps required to do disk transfers; but, in addition, performs continuing self-testing of the individual controller channel on a time-available basis to assure that the components are continuing to work properly. Should such self-check indicate a potential problem, the CPU 60 then conducts further testing of the subject disk controller channel 46 to see if an on-line "hot" spare disk drive 16 or disk controller channel 46 should be employed while the malfunctioning unit is flagged for work on the display 44 and removed from use temporarily.

By putting the CPU 60, by way of micro-processor 58, in complete charge of transfers between the memories 48 and the cache memory 56 the problem of dealing with bus arbitration on the XBUS 64 is also eliminated. That, of course, has many beneficial effects such as increased reliability, increased throughput, and simplicity of implementation.

To get a better idea of the operation of the disk platform 28 of this invention, we will now turn to Figures 4 and 5 with particularity. Figure 4 shows further details of the interface and driver unit generally labelled as 42 while Figure 5 shows further details of one of the disk controller channels 46. With reference first to Figure 4, there is an XBUS driver 66 connected to the XBUS 64 with a connector 68. There is also a host interface driver 70 (SCSI in the tested embodiment) connected into line 20 back to the host computer 10 by a connector 72. As with the other elements, there is also a local micro-processor 74 to control the operation of the elements of the interface and driver unit 42. The micro-processor 74 interfaces with a display driver 76 (which is

connected to the display 44) and a serial interface driver 78 (which is connected to the serial interface on line 18). All the driver elements are well known to those skilled in the art and will be chosen according to the type of device with which they must interface. The micro-processor 74 is also connected to the CPU bus 52 with connector 80. The heart of the interface and driver unit 42 and most important part thereof is a pair of unidirectional FIFOs 82 and 84. Under the control of the local micro-processor 74, FIFO 82 receives and transmits data from the XBUS 66 to the host computer 10. Similarly, FIFO 84 receives and transmits requests and data from the host computer 10 to the XBUS 66. Thus, bi-directional transmissions can take place between the XBUS 66 and the host computer 10. This is another feature of this embodiment which improves the overall throughput of the system 22.

The disk controller channels 46 depicted in Figure 5 also includes an XBUS driver 66 and a disk drive interface driver 92 connected to the associated disk drive 16 with their associated connectors 68, 94. Similarly, the local micro-processor 50 is connected to the CPU bus 52 with a connector 80. In addition to the data buffer memory 48, there is a buffer address register 86 which controls the locations in the memory 48 which are accessed and a data traffic semaphore 88 which operates in a manner readily apparent to those skilled in the art to control access to and from the memory 48 under the control of the micro-processor 58. Thus, it is the data traffic semaphore 88 which actually inserts the addresses into the buffer address register 86. The data traffic semaphore 88 must be present to prevent simultaneous access to the memory 48 by both the XBUS 64 (and elements connected therethrough) and the host computer 10. Such devices are well known and employed in the computing art for that purpose as any attempt to simultaneously read from and write into a memory location can cause irreversible errors.

The near line archival storage channel 100 is controlled in the same manner as disk controller channel 46 thru microprocessor 50 and cache/buffer memory 48 and contains the logic to control by way of control bus 101 the near line archival storage 103 and its individual elements 104, 105 and 106 to read and write data by way of data bus 102. Data read from near line archival storage 103 is held in cache memory 56 or on disk drives 16 and is accessed by the host computer with sector numbers beyond the physical limits of disk drives 16 creating a virtually boundless storage capacity.

Having thus described the construction and operation of the system 22 in general, a more specific example of its unique mode of operation will now be described with reference to Figure 6. For simplicity, Figure 6 depicts in simplified form only the cache/buffer memories 48 in the channels and the cache memory 56 as connected by the XBUS 64. Assume that a request has been made by the host computer 10 to read data. The disk platform 28, of course, knows (or can determine) the location of the data in the disk drives 16 through its above-described access to the tables 24 in the host computer 10. According to the fault tolerant techniques described in the parent application, the data (and its parity bits) are spread across the disk drives 16. From the contents of its private memory 62, the logic in the CPU 60 knows the present contents of the cache memory 56. Anything that is already in the cache memory 56 will not be re-read, of course, which is the usual function of any cache memory (i.e. to eliminate redundant and unnecessary disk accesses). The CPU 60 then issues high level requests to the various disk controller channels 46 to have them retrieve the elements of the data from their locations on the disk drives. The requests also go to the cache memory & control unit 54 so that it knows what is going on. From there on, the collection of the data and its transfer to the host computer 10 is under the control of the micro-

processor 58 in the cache memory & control unit 54. The micro-processor 58 assigns available buffer space (as indicated by the dashed box 90) in which to receive the data of the request. The data segments are asynchronously brought into the buffer memories 48 under the control of the micro-processors 50 as a function of the originally-issued high level commands from the CPU 60. As the data segments are received, the micro-processor 58 is advised by the micro-processors 50 over the XBUS 64. The micro-processor 58 then asynchronously transfers the data segments into their appropriate location within the assigned buffer space 90. When the entire data of the request is in the assigned buffer space 90, the micro-processor 58 transfers it to the host computer 10 through the FIFO 82 described above. A write operation, of course, operated in much the same manner, except that data flow is in the opposite direction.

While only shown in simplified representation in Figure 6, it may be appreciated therefrom and from a consideration of the elements and their method of operation as described above that the single cache memory 56 of substantial size as employed in this embodiment effect a vast improvement in simplicity and efficiency of operation (i.e. speed). Because of its size, the cache memory 56 will actually self-optimize the data it retains in cache over time so as to minimize the number of data transfers required with the disk drives. In this regard, it is important to recognize that the parity bits associated with data are contained in separately transferrable locations on the disk drives 16. Thus, over the course of time the most used data and their parity bits will be virtually permanent residents of the cache memory 56 and will only be written to the disk drives 16 for back-up purposes on a time-available basis in a background mode by the micro-processor 58. The impact of this on the overall performance of the system 22 should not be overlooked or minimized. Consider, if there are

eight disk drives 16 containing the individual data bits of data and a ninth disk drive 16 containing the parity bits associated therewith, if the parity bits are contained in an unused portion of the cache memory 56 and only written to the ninth disk drive 16 when the eight disk drives 16 are not otherwise involved in an actual data transfer, disk access time to access data is reduced by one-ninth or eleven per-cent. The savings in time could be even more substantial in an instance where one of the eight disk drives 16 containing the data bits is malfunctioning and the parity bits are required to correct any errors in the data stream.

Thus, it can be seen from the foregoing description that the embodiments described herein provide substantial and novel improvements to the basic computer storage system of the parent application of which this is a continuation-in-part.

As with the parent application, those skilled in the art will readily recognize and appreciate that the use of disk drives as employed hereinbefore by way of example is a convenience only and that the techniques described herein could be, and are, applicable to other storage devices presently available or under development such as, for example, CD-ROMs. It is the inventors' intent and hope that the breadth accorded the inventions described in this application and the claims appended hereto be in keeping with the scope and spirit of the invention and not be limited in any way because of the use of the term disk drive in the specification and/or claims.

Wherefore, having thus described our invention, what is claimed is:

1. In a computer system including a host computer connected to access a plurality of data storage units and including an operating system having tables of file locations of data stored on said data storage units, the improvement comprising:

a) data storage unit access control means connected between the host computer and the plurality of data storage units, for receiving read and write requests from the host computer, for reading and writing data from and to the plurality of data storage units, and for transferring data to and from the host computer;

b) host computer link means, connected between said data storage access control means and said host computer, for providing said data storage access control means with said read and write requests from said host computer, and for transferring data to and from said host computer;

c) host computer operating system communications link means, connected between said data storage unit access control means and a console privilege interface of the host computer operating system, for obtaining and providing said data storage unit access control means with data file manipulation commands and associated data files storage locations, said data file manipulation commands for directing said data storage unit to manipulate predetermined data stored in said data file storage location associated with said data file manipulation commands; and,

d) computational logic means contained in said data storage unit access control means and responsive to at least said data file manipulation commands and associated data files storage locations received over said host computer operating systemcommunications link means,

for providing data storage unit command signals directing said data storage unit to manipulate said predetermined data stored in said data file storage locations associated with said data file manipulation commands asynchronously with and independent of said read and write requests from said host computer over said host computer link means, whereby the host operating system provides transparency about data file manipulation commands to user programs operating within the host computer and reading and writing data to and from the plurality of data storage unit over said host computer link means.

2. The improvement to a computer system of claim 1 wherein:

said computational logic means includes learning logic so that once the location of a requested data on the plurality of data storage units has been obtained using said communications link means, it is known and remembered.

3. The improvement to a computer system of claim 1 wherein:

a) the computer system includes near line archival storage apparatus of an automated variety wherein off-line storage media are loaded for reading and writing on request by automated mechanisms; and,

b) said computational logic means includes logic for using said communications link means to obtain file location information about data on said near line storage apparatus.

4. The improvement to a computer system of claim 3 wherein additionally:

said computational logic means includes learning logic so that once the location of a requested data on said near line storage apparatus has been obtained using said communications link means, it is known and remembered.

5. The improvement to a computer system of claim 4 wherein additionally:

said learning logic of said computational logic means is of a self learning variety which learns through experience so that if a particular file from said near line storage apparatus is repeatedly used at a given time said learning logic learns that from experience and begin loading said file from said near line archival storage apparatus in anticipation of an expected request so that when said request is made, said file is already read in and available for use.

6. The improvement to a computer system of claim 1 wherein:

said computational logic means includes testing logic for testing the plurality of data storage units in a background mode on a time-available basis.

7. The improvement to a computer system of claim 1 wherein said data storage unit access control means comprises:

a) a plurality of data storage unit controller channel means connected to respective ones of the plurality of data storage units for controlling transfers of data to and from an associated one of the plurality of data storage units in response to received high level commands, each of said plurality of data storage unit

controller channel means including a cache/buffer memory and a micro-processor unit for controlling said transfers of data and status;

b) interface and driver unit means for interfacing with said communications link means and the host computer;

c) a central cache memory;

d) cache memory control means for controlling transfers of data from said cache/buffer memory of said plurality of data storage unit controller channel means to said cache memory and from said cache memory to said cache/buffer memory of said plurality of data storage unit controller channel means and from said cache memory to the host computer through said interface and driver unit means;

e) central processing unit means for managing the use of said cache memory by requesting data transfers only with respect to ones of the plurality of data storage units where data associated therewith is not presently in said cache memory and for sending high level commands to said plurality of data storage unit controller channel means to effect data transfers thereby;

f) first (data) bus means interconnecting said plurality of data storage unit controller cache/buffer memories, said interface and driver unit cache/buffer memory and said cache memory for the transfer of information therebetween; and,

g) second (information and commands) bus means interconnecting said plurality of data storage unit controller channel means, said interface and driver unit means, said cache memory control means, and said central processing unit means for the transfer of control information therebetween.

8. The improvement to a computer system of claim 7 wherein said data storage unit access control means includes:

a) an alphanumeric display; and,

b) driver means for driving said display.

9. The improvement to a computer system of claim 7 wherein said interface and driver unit means comprises:

a) first bus driver means for connecting to and interfacing with said first bus;

b) host computer interface driver means for connecting to and interfacing with the host computer;

c) communications link interface means for connecting to and interfacing with said communications link means;

d) a first unidirectional FIFO connected between said first bus driver means and said host computer interface driver means for receiving inputs from said first bus intended for the host computer;

e) a second unidirectional FIFO connected between said host computer interface driver means and said first bus driver means for receiving inputs from the host computer intended for said first bus;

f) local micro-processor means for controlling said first unidirectional FIFO, said second unidirectional FIFO, and said host computer interface driver means; and,

g) connector means for connecting said local micro-processor means to said second bus.

10. The improvement to a computer system of claim 7 wherein each of said plurality of data storage unit controller channel means comprises:

a) first bus driver means for connecting to and interfacing with said first bus;

b) data storage unit interface driver means for connecting to and interfacing with an associated one of the plurality of data storage units;

c) a data buffer;

d) data semaphore means for preventing simultaneous access to said data buffer by said first bus driver means and said data storage unit interface driver means;

e) buffer address register means for providing a current active address into said data buffer;

f) local micro-processor means for controlling said data semaphore means and said data storage unit interface driver means; and,

g) connector means for connecting said local micro-processor means to said second bus.

11. In a computer system including a host computer connected to access a plurality of disk drives to provide an a synchronously operating, improved disk drive access control apparatus connected between the host computer and the plurality of disk drives for receiving read and write requests from the host computer, for reading and writing data from and to the plurality of disk drives, and for transferring data to and from the host computer, said disk drive access control apparatus comprising:

a) a plurality of disk drive controller channel means connected to respective ones of the plurality of disk drives for controlling transfers of data to and from an associated one of the plurality of disk drives in response to received high level commands, each of said plurality of disk drive controller channel means

including a cache/buffer memory and a micro-processor unit for controlling said transfers of data;

b) interface and driver unit means for interfacing with a host computer operating system communications link means connected between said disk drive access control apparatus and a console privilege interface of a host computer operating system and a host computer link means connected between said disk drive access control apparatus and the host computer;

c) a central cache memory;

d) cache memory control means for controlling transfers of data from said cache/buffer memory of said plurality of disk drive controller channel means to said cache memory and from said cache memory to said cache/buffer memory of said plurality of disk drive controller channel means and from said cache memory to the host computer through said interface and driver unit means;

e) central processing unit means for managing the use of said cache memory by requesting data transfers only with respect to ones of the plurality of disk drives where data associated therewith is not presently in said cache memory, for sending high level commands to said plurality of disk drive controller channel means to effect data transfers thereby, and for receiving over said host computer operating system communication link means, data file manipulation commands and associated data file storage locations, for providing data storage unit command signals controlling manipulation of predetermined data stored in said data file storage locations associated with said data file manipulation commands asynchronously with and independent of read and write requests from said host computer over said host computer link means;

f) first (data) bus means interconnecting said plurality of data storage unit controller cache/buffer memories, said interface and driver unit means, said cache/buffer memory and said cache memory, for the transfer of information therebetween; and,

g) second (information and commands) bus means interconnecting said plurality of data storage unit controller channel means, said interface and driver unit means, said cache memory control means, and said central processing unit means for the transfer of control information therebetween.

12. The improvement to a computer system of claim 11 wherein said disk drive access control means includes: ;

a) an alphanumeric display; and,

b) driver means for driving said display.

13. The improvement to a computer system of claim 11 wherein said interface and driver unit means comprises:

a) first bus driver means for connecting to and interfacing with said first bus;

b) host computer link interface driver means, for connecting to and interfacing with the host computer;

c) host computer operating system communications link interface driver means, for connecting and interfacing with said host computer console privilege interface;

d) a first unidirectional FIFO connected between said first bus driver means and said host computer interface driver means, for receiving inputs from said first bus intended for the host computer;

e) a second unidirectional FIFO connected between said host computer interface driver means and said

first bus driver means for receiving inputs from the host computer intended for said first bus;

f) local micro-processor means for controlling said first unidirectional FIFO, said second unidirectional FIFO, and said host computer interface driver means; and,

g) connector means for connecting said local micro-processor means to said second bus.

14. The improvement to a computer system of claim 11 wherein each of said plurality of disk drive controller channel means comprises:

a) first bus driver means for connecting to and interfacing with said first bus;

b) disk drive interface driver means for connecting to and interfacing with an associated one of the plurality of disk drives;

c) a data buffer;

d) data semaphore means for preventing simultaneous access to said data buffer by said first bus driver means and said disk drive interface driver means;

e) buffer address register means for providing a current active address into said data buffer;

f) local micro-processor means for controlling said data semaphore means and said disk drive interface driver means; and,

g) connector means for connecting said local micro-processor means to said second bus.

15. Disk drive access control apparatus for connection between a host computer and a plurality of disk drives to provide an a synchronously operating, storage system to receive read and write requests from the host computer, data file manipulation commands, read and write

data from and to the plurality of disk drives, and transfer data to and from the host computer, said apparatus comprising:

a) a plurality of disk drive controller channels connected to respective ones of the plurality of disk drives and controlling transfers of data to and from an associated one of the plurality of disk drives in response to received high level commands, each of said plurality of disk drive controller channels including a cache/buffer memory and a micro-processor unit for controlling said transfers of data;

b) an interface and driver unit interfacing with a host computer operating system communications link means connected between said disk drive control apparatus and a console privilege interface of a host computer operating system, and a host computer link connected between said disk drive access control apparatus and said host computer;

c) a central cache memory;

d) cache memory control logic controlling transfers of data from said cache/buffer memory of said plurality of disk drive controller channels to said cache memory and from said cache memory to said cache/buffer memory of said plurality of disk drive controller channels and from said cache memory to the host computer through said interface and driver unit;

e) a central processing unit managing the use of said cache memory by requesting data transfers only with respect to ones of the plurality of disk drives where data associated therewith is not presently in said cache memory and by sending high level commands to said plurality of disk drive controller channels to effect data transfers thereby, and for receiving over said host

computer operating system communication link means, data file manipulation commands and associated data file storage locations, for providing data storage unit command signals controlling manipulation of predetermined data stored in said data file storage locations associated with said data file manipulation commands asynchronously with and independent of read and write requests from said host computer over said host computer link means;;

f) a first (data) bus means interconnecting said plurality of data storage unit controller cache/buffer memories, said interface and driver unit cache/buffer memory and said cache memory for the transfer of information therebetween; and,

g) second (information and commands) bus means interconnecting said plurality of data storage unit controller channel means, said interface and driver unit means, said cache memory control means, and said central processing unit means for the transfer of control information therebetween.

16. The disk drive access control apparatus of claim 15 wherein said disk drive access control logic includes:

a) an alphanumeric display; and,

b) driver means for driving said display.

17. The disk drive access control apparatus of claim 15 wherein said interface and driver unit comprises:

a) first bus driver means for connecting to and interfacing with said first bus;

b) host computer link interface driver means for connecting to and interfacing with the host computer;

c) host computer operating system communication link interface driver means, for connecting to and interfacing with said host computer operating system console privilege interface;

d) a first unidirectional FIFO connected between said first bus driver means and said host computer interface driver means for receiving inputs from said first bus intended for the host computer;

e) a second unidirectional FIFO connected between said host computer interface driver means and said first bus driver means for receiving inputs from the host computer intended for said first bus;

f) local micro-processor means for controlling said first unidirectional FIFO, said second unidirectional FIFO, and said host computer interface driver means; and,

g) connector means for connecting said local micro-processor means to said second bus.

18. The disk drive access control apparatus of claim 15 wherein each of said plurality of disk drive controller channels comprises:

a) first bus driver means for connecting to and interfacing with said first bus;

b) disk drive interface driver means for connecting to and interfacing with an associated one of the plurality of disk drives;

c) a data buffer;

d) data semaphore means for preventing simultaneous access to said data buffer by said first bus driver means and said disk drive interface driver means;

e) buffer address register means for providing a current active address into said data buffer;

f) local micro-processor means for controlling said data semaphore means and said disk drive interface driver means; and,

g) connector means for connecting said local micro-processor means to said second bus.

19. The disk drive access control apparatus of claim 15 and additionally comprising:

computational logic means for using said communications link means to obtain file location information about a data manipulation request from the host computer whereby whereby said host operating system provides transparency about said data file manipulation commands to user programs operating within the host computer and reading and writing data to and from the plurality of disk drives over said host computer link means.

20. The disk drive access control apparatus of claim 19 wherein:

said computational logic means includes learning logic so that once the location of a requested data on the plurality of disk drives has been obtained using said communications link means, it is known and remembered.

21. The disk drive access control apparatus of claim 19 wherein:

a) a computer system including the host computer and disk drives further includes near line archival storage apparatus of an automated variety wherein

off-line storage media are loaded for reading and writing on request by automated mechanisms;

b) said computational logic means includes logic for using said communications link means to obtain location information about files on said near line archival storage apparatus; and

c) said computational logic means managing the near line archival storage apparatus to create a virtual array, of accessible data including the capacity of the entire near line archival storage apparatus, with the appearance of a disk drive of boundless capacity.

22. The disk drive access control apparatus of claim 21 wherein additionally:

said computational logic means includes learning logic so that once the location of a requested data on said near line storage apparatus has been obtained using said communications link means, it is known and remembered.

23. The disk drive access control apparatus of claim 22 wherein additionally:

said learning logic of said computational logic means is of a self learning variety which learns through experience so that if a particular file from said near line storage apparatus is repeatedly used at a given time said learning logic learns that from experience and begin loading said file from said near line storage apparatus in anticipation of an expected request so that when said request is made, said file is already read in and available for use.

24. The improvement to a computer system of claim 19 wherein:

said computational logic means includes testing logic for testing the plurality of disk drives in a background mode on a time-available basis.

25. A method of operating the disk drive access control apparatus of claim 17, wherein said apparatus performs the steps of:

a) controlling transfers of data from the cache/buffer memory of the plurality of disk drive controller channels to the cache memory and from the cache memory to the cache/buffer memory of the plurality of disk drive controller channels and from the cache memory to the host computer through the interface and driver unit; and,

b) managing the use of the cache memory by requesting data transfers only with respect to ones of the plurality of disk drives where data associated therewith is not presently in the cache memory and sending high level commands to the plurality of disk drive controller channels to effect data transfers thereby.

26. The method of claim 25 and additionally including the steps of:

a) providing a first unidirectional FIFO connected between the first bus driver means and the host computer interface driver means for receiving inputs from the first bus intended for the host computer;

b) providing a second unidirectional FIFO connected between the host computer interface driver means and the first bus driver means for receiving inputs from the host computer intended for the first bus; and,

c) using a local micro-processor to control the first unidirectional FIFO and the second unidirectional FIFO.

27. The method of claim 25 wherein said apparatus additionally performs the step of:

employing a data semaphore to prevent simultaneous access to the data buffer by a first bus driver and a disk drive interface driver.

28. The method of claim 25 and additionally including the steps of:

a) providing a communications link to the host computer; and,

b) employing the communications link to obtain the location of files on the plurality of disk drives from the host computer.

29. The method of claim 25 wherein sid apparatus additionally performs the steps of:

a) saving a history of data accesses by the host computer;

b) anticipating data accesses from the history of data accesses and retrieving the data associated with the accesses from the disk drives before an actual request therefor is made by the host; and

c) managing the near line archival storage apparatus to create a virtual array, of accessible data including the capacity of the entire near line archival storage apparatus, with the appearance of a disk drive of boundless capacity.

30. The method of claim 25 where a computer system including the host computer and disk drives further includes near line storage apparatus of an automated variety wherein off-line storage media are loaded for reading and writing on request by automated mechanisms so as to avoid the necessity of operator intervention and wherein said apparatus additionally performs the steps of:

employing the host computer operating system communications link to obtain the location of data on the near line storage apparatus from the host computer operating system.

31. The method of claim 30 wherein said apparatus additionally performs the steps of:

a) saving a history of data accesses by the host computer;

b) anticipating data accesses from the history and retrieving the data associated with the accesses from the near line storage apparatus before an actual request therefor is made by the host computer; and

c) managing the near line archival storage apparatus to create a virtual array, of accessible data including the capacity of the entire near line archival storage apparatus, with the appearance of a disk drive of boundless capacity.

2087162



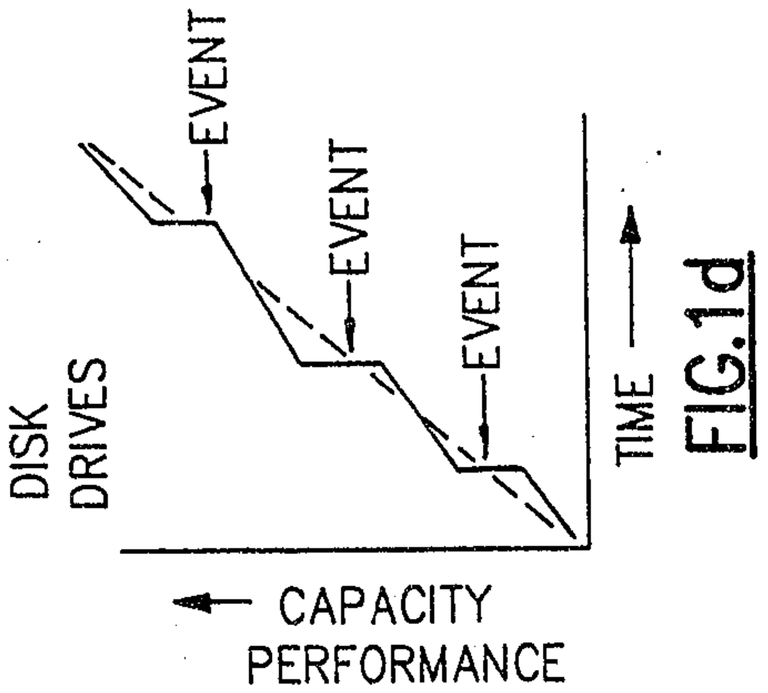



FIG.1d

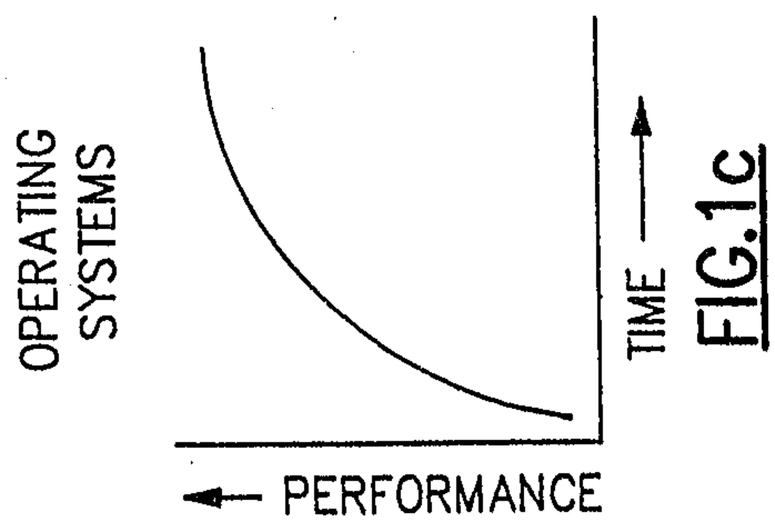



FIG.1c

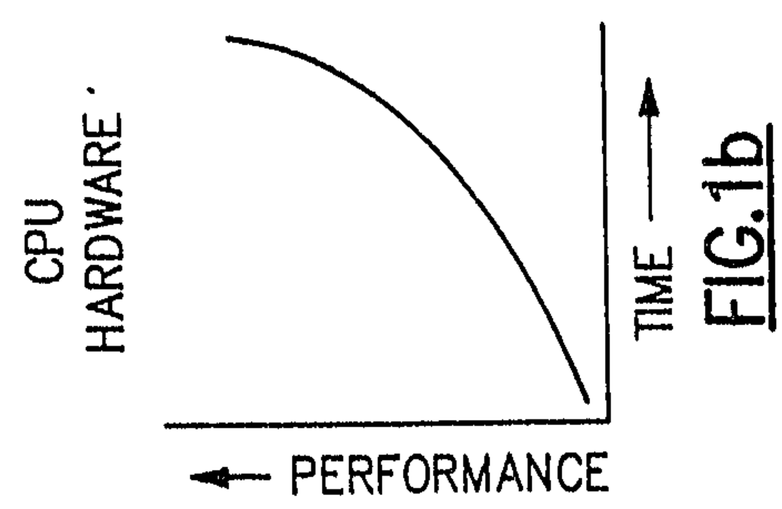



FIG.1b

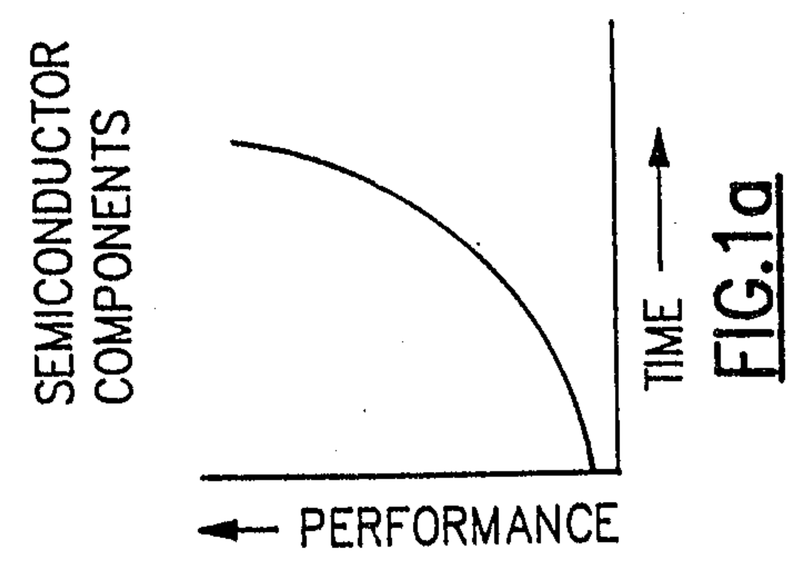



FIG.1a

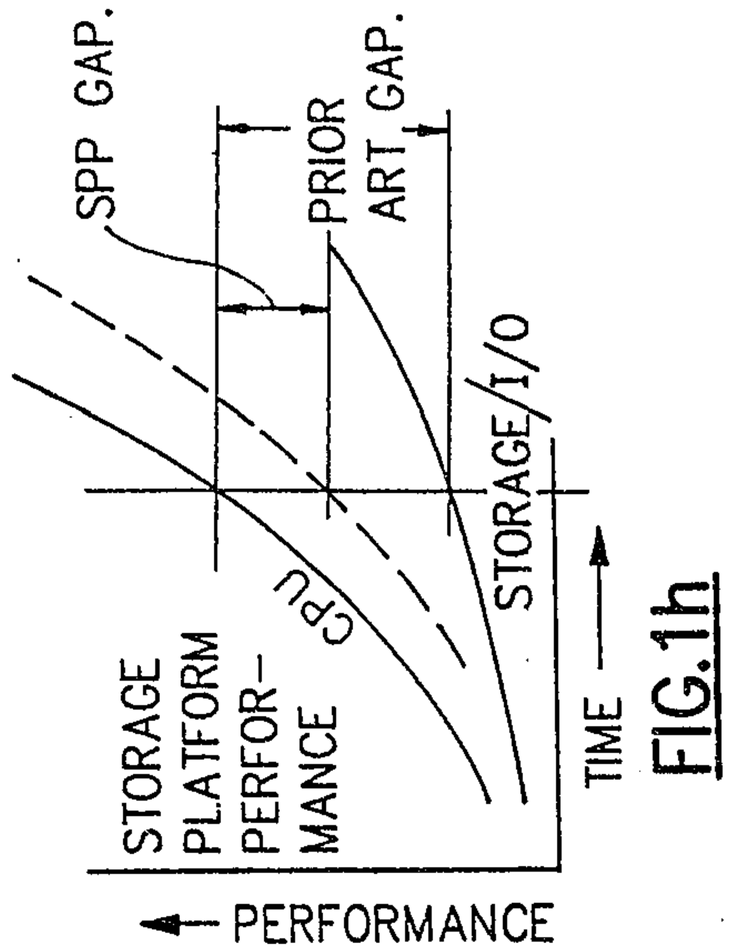



FIG.1h

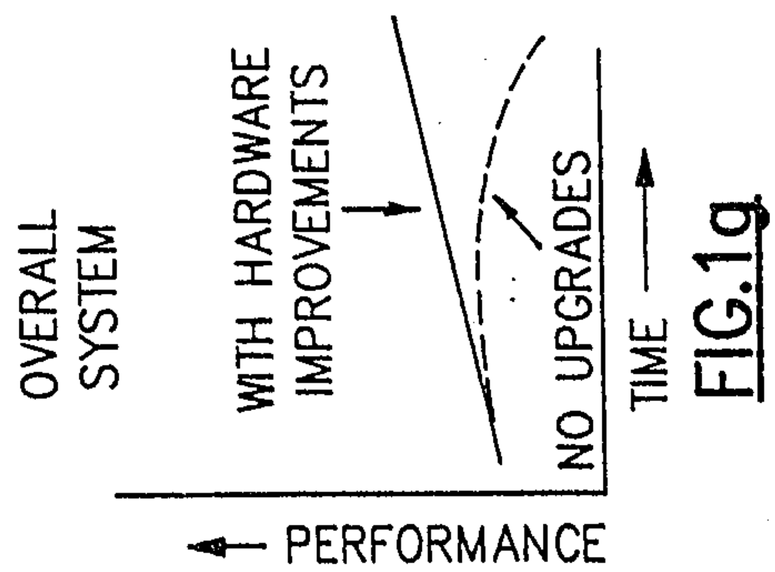



FIG.1g

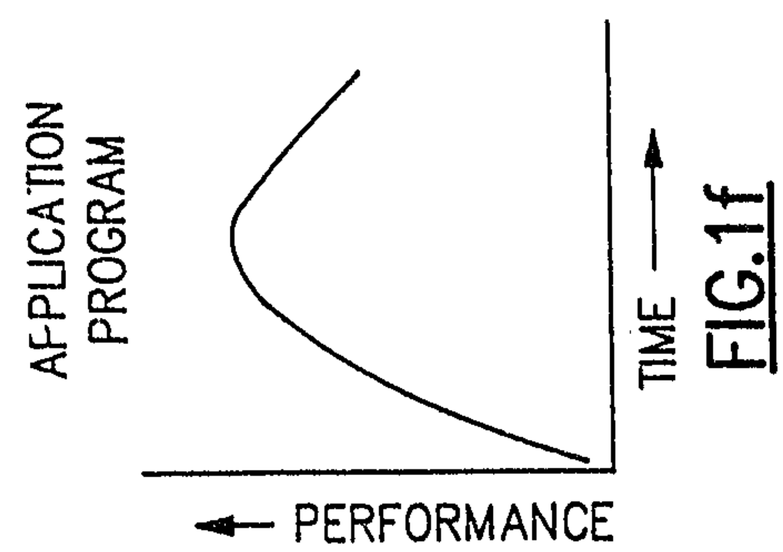



FIG.1f

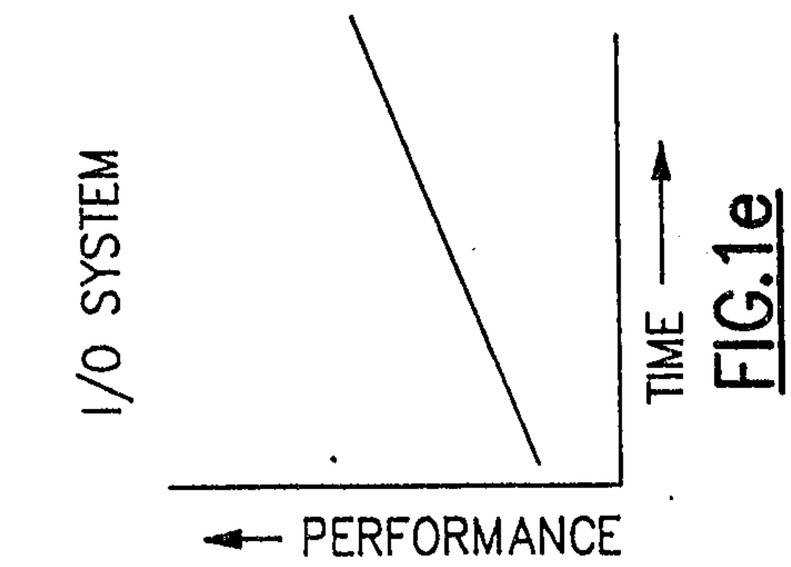
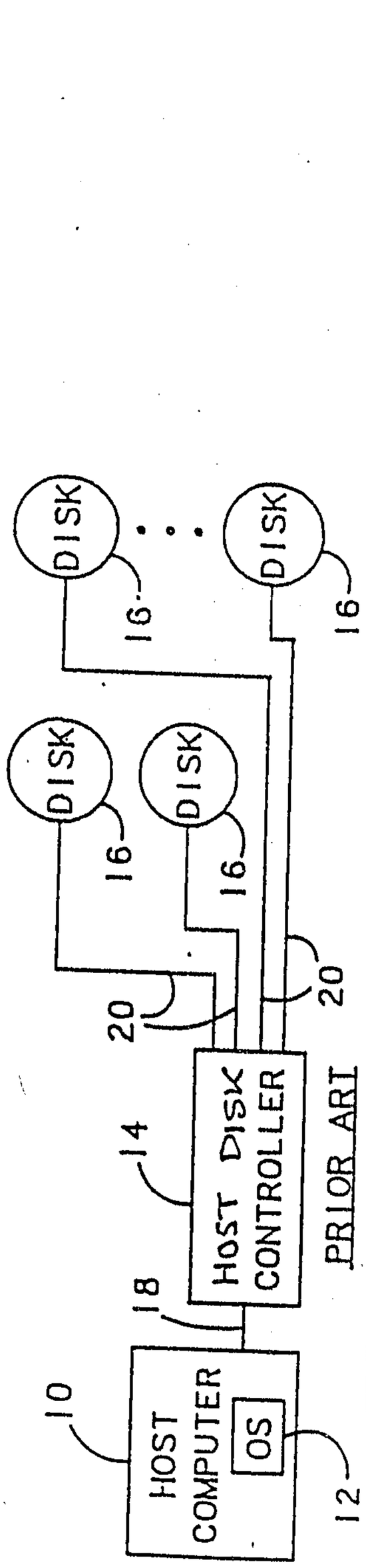



FIG.1e




PRIOR ART

FIG. 1(b)

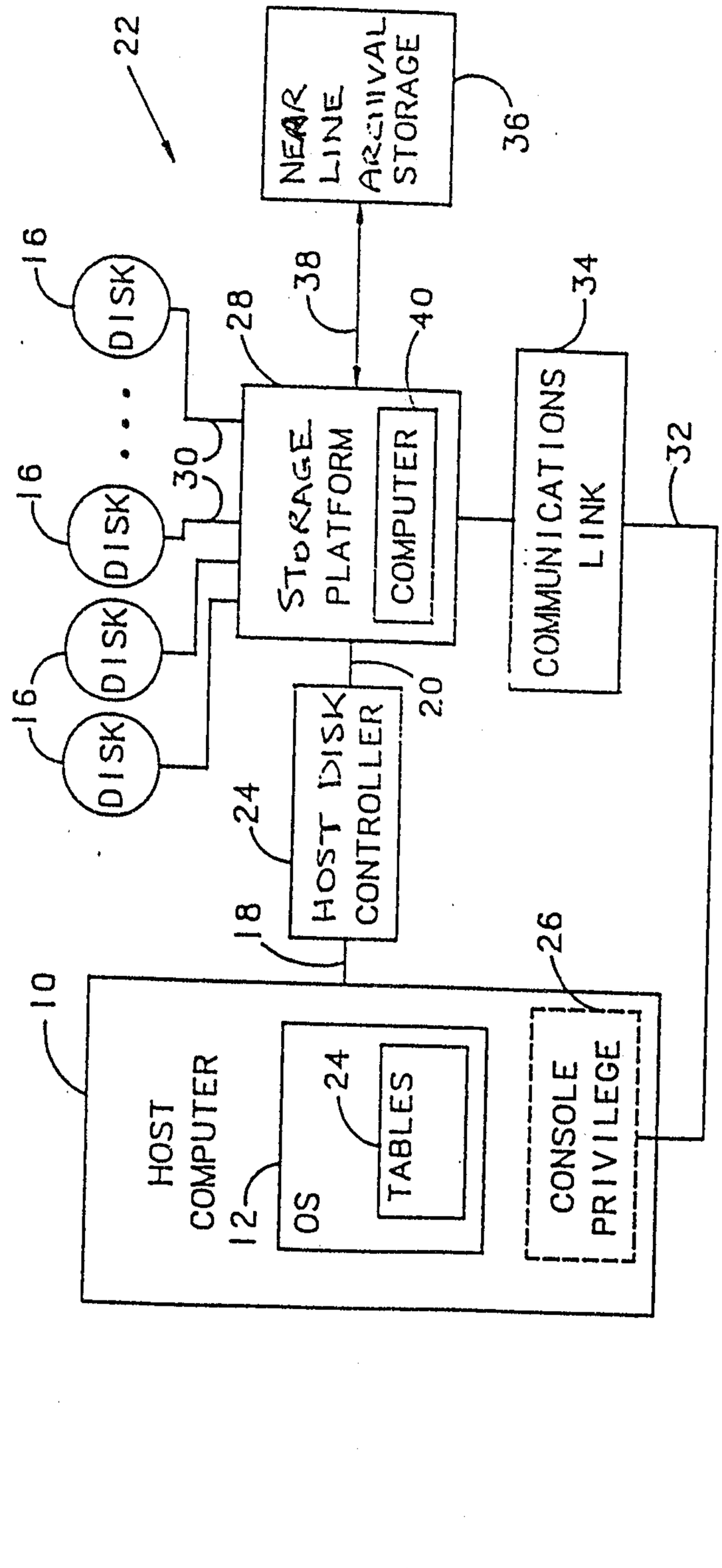



FIG. 2

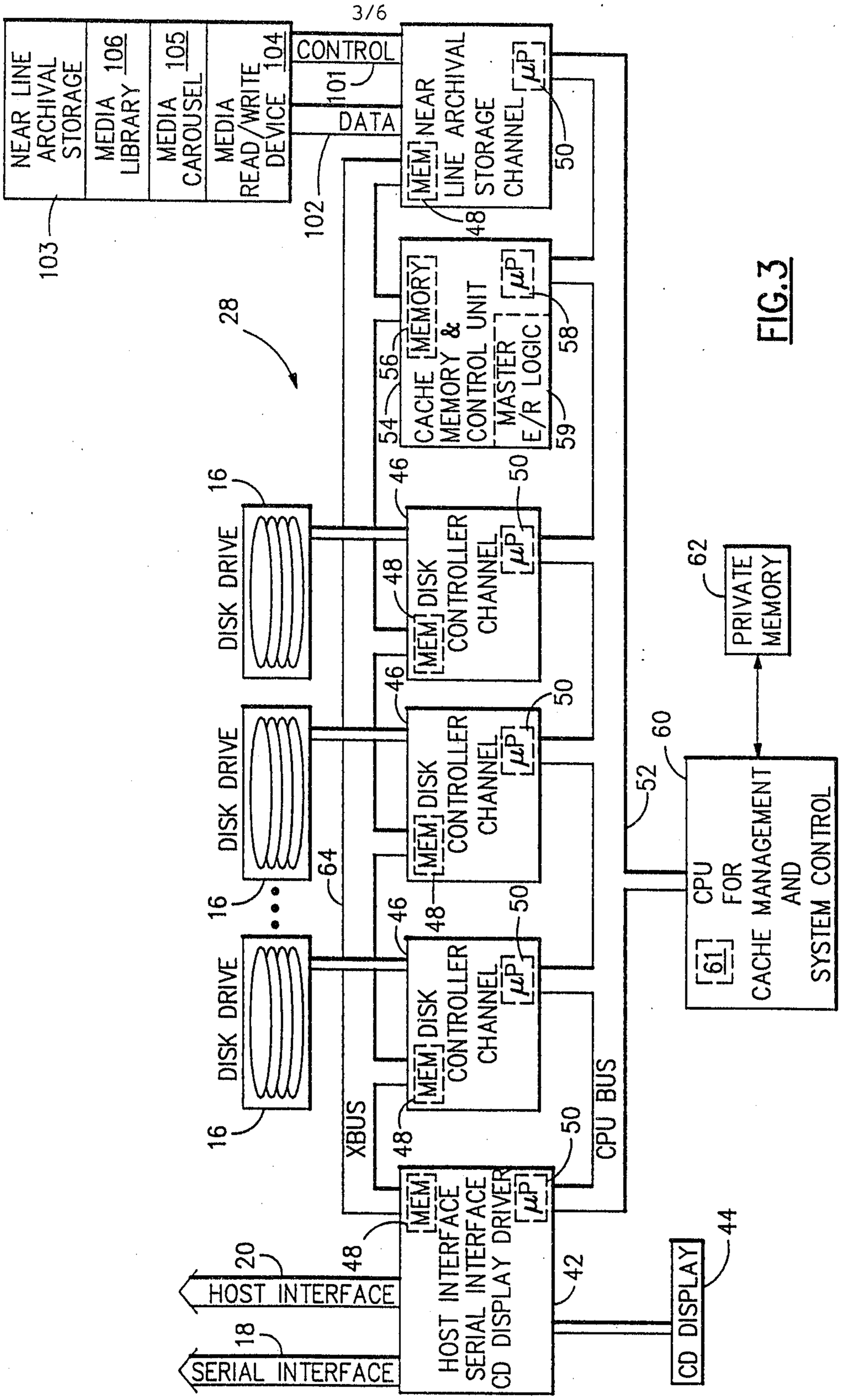

NEAR LINE ARCHIVAL STORAGE
MEDIA LIBRARY 106
MEDIA CAROUSEL 105
MEDIA READ/WRITE DEVICE 104
CONTROL
101
DATA
102
103
[MEM] NEAR LINE ARCHIVAL STORAGE CHANNEL [μP]
48
50
28
16
DISK DRIVE
[MEM] DISK CONTROLLER CHANNEL [μP]
46
54
56
CACHE [MEMORY] MEMORY & CONTROL UNIT
MASTER E/R LOGIC [μP]
58
59
64
XBUS
CPU BUS
[MEM] HOST INTERFACE SERIAL INTERFACE CD DISPLAY DRIVER [μP]
20
HOST INTERFACE
18
SERIAL INTERFACE
42
CD DISPLAY
44
52
60
62
PRIVATE MEMORY
[61] CPU FOR CACHE MANAGEMENT AND SYSTEM CONTROL

FIG. 3

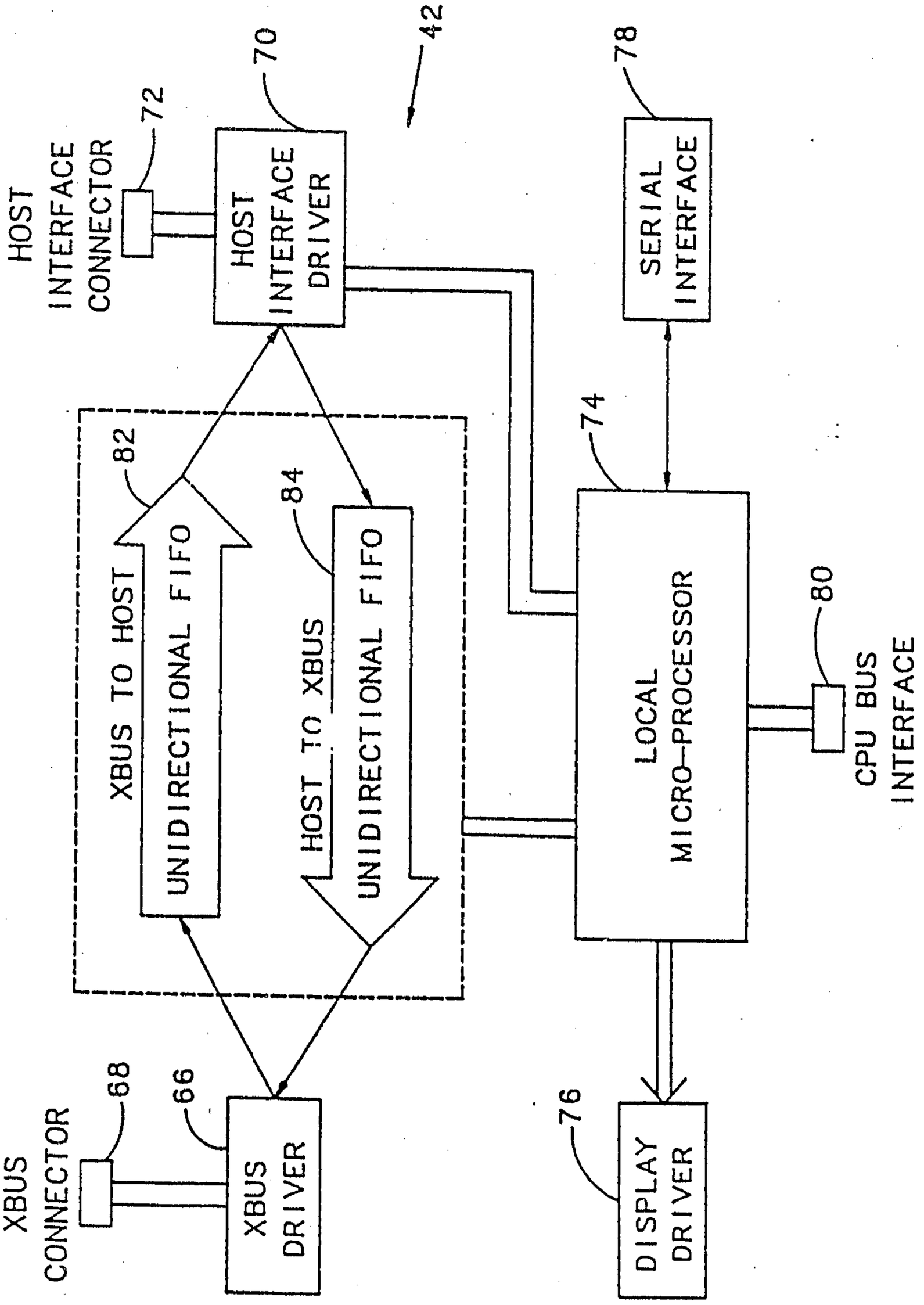

HOST INTERFACE CONNECTOR
72
HOST INTERFACE DRIVER
70
SERIAL INTERFACE
78
XBUS CONNECTOR
68
XBUS DRIVER
66
XBUS TO HOST UNIDIRECTIONAL FIFO
82
HOST TO XBUS UNIDIRECTIONAL FIFO
84
LOCAL MICRO-PROCESSOR
74
DISPLAY DRIVER
76
CPU BUS INTERFACE
80

FIG. 4

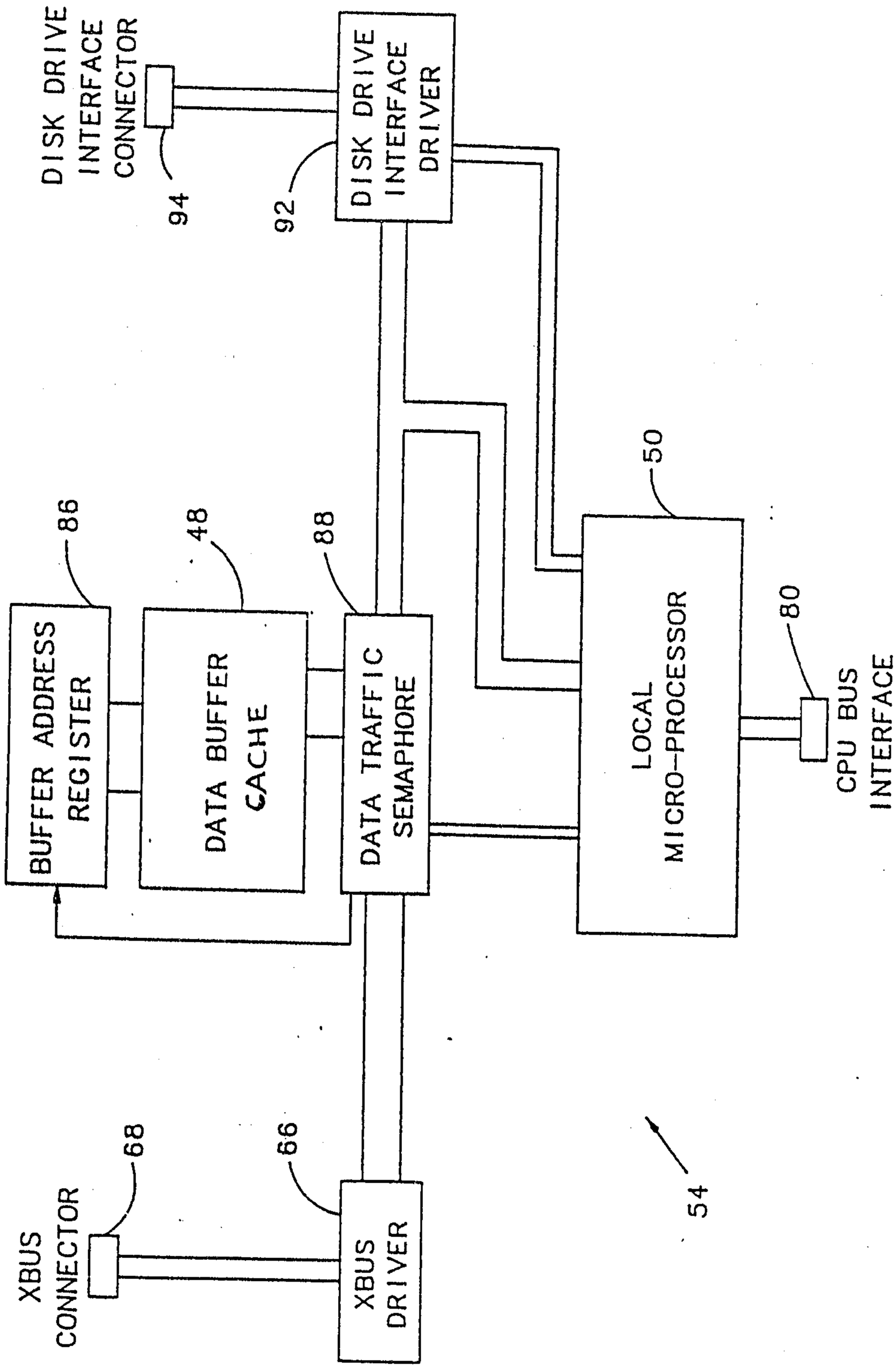

DISK DRIVE INTERFACE CONNECTOR
94
92
DISK DRIVE INTERFACE DRIVER
86
48
88
50
BUFFER ADDRESS REGISTER
DATA BUFFER CACHE
DATA TRAFFIC SEMAPHORE
LOCAL MICRO-PROCESSOR
80
CPU BUS INTERFACE
68
66
XBUS CONNECTOR
XBUS DRIVER
54

FIG. 5

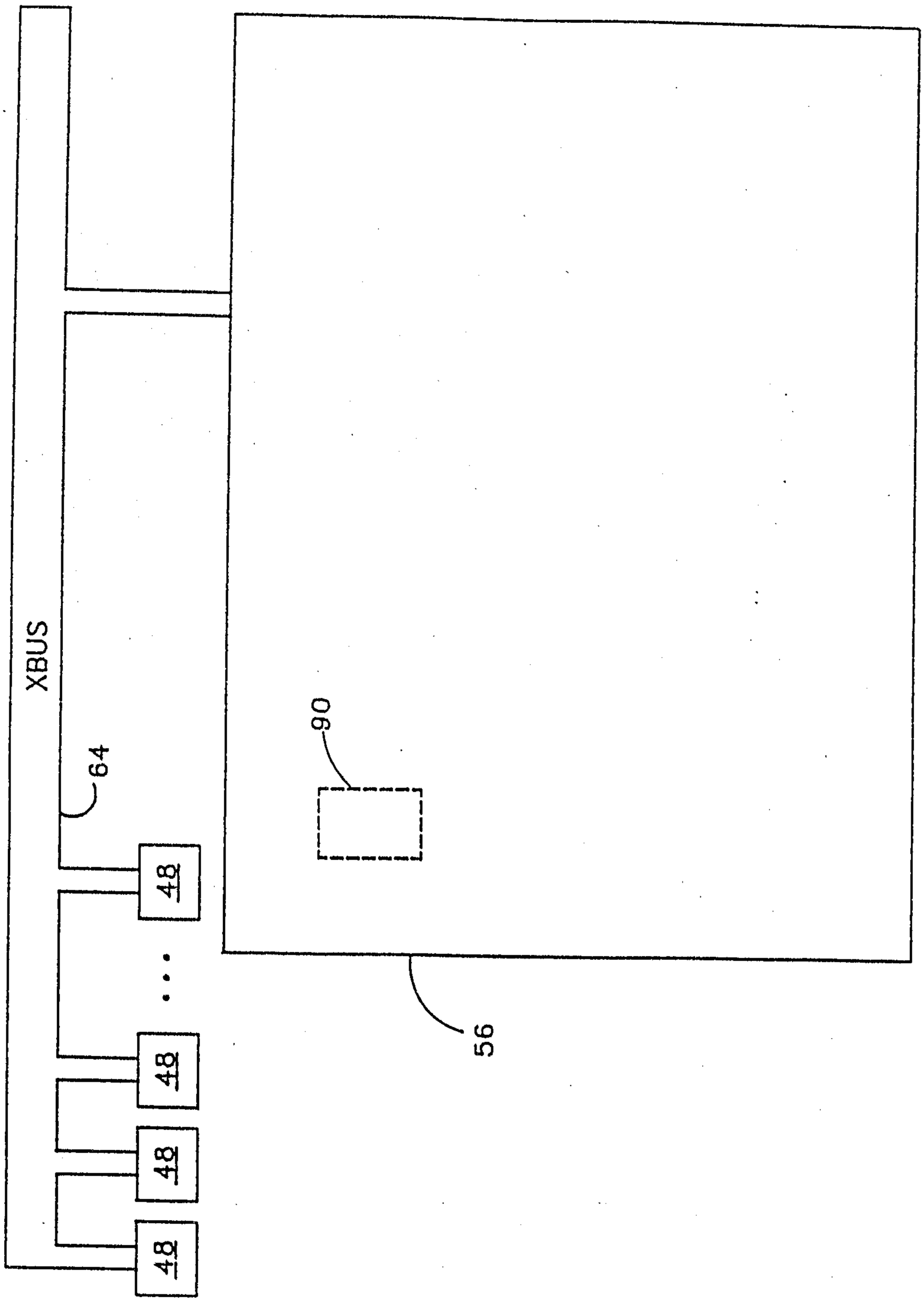
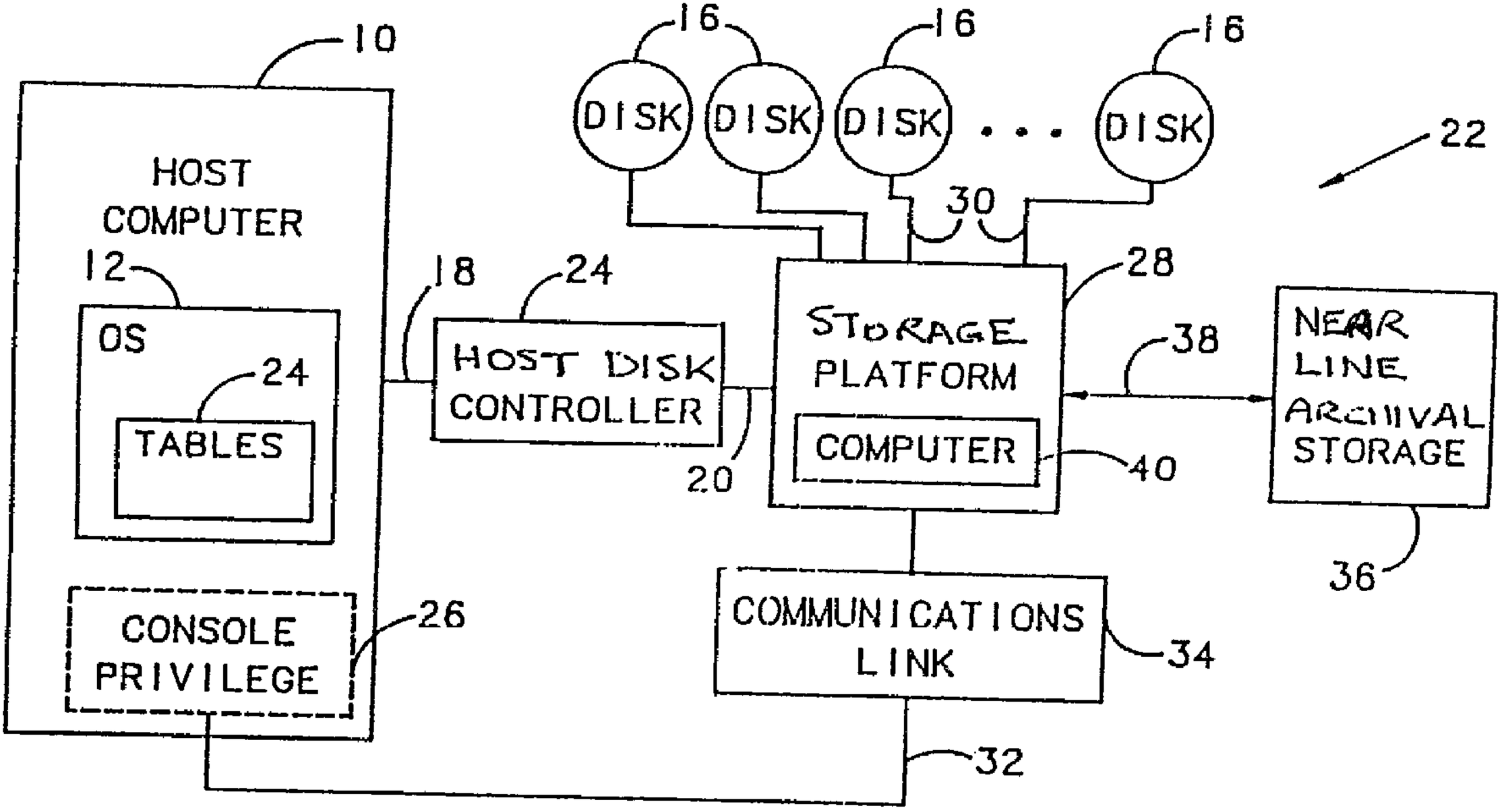



FIG. 6


10
HOST COMPUTER
12
OS
24
TABLES
CONSOLE PRIVILEGE
26
16
DISK
DISK
DISK
DISK
22
30
18
24
HOST DISK CONTROLLER
20
STORAGE PLATFORM
28
COMPUTER
40
38
NEAR LINE ARCHIVAL STORAGE
36
COMMUNICATIONS LINK
34
32