



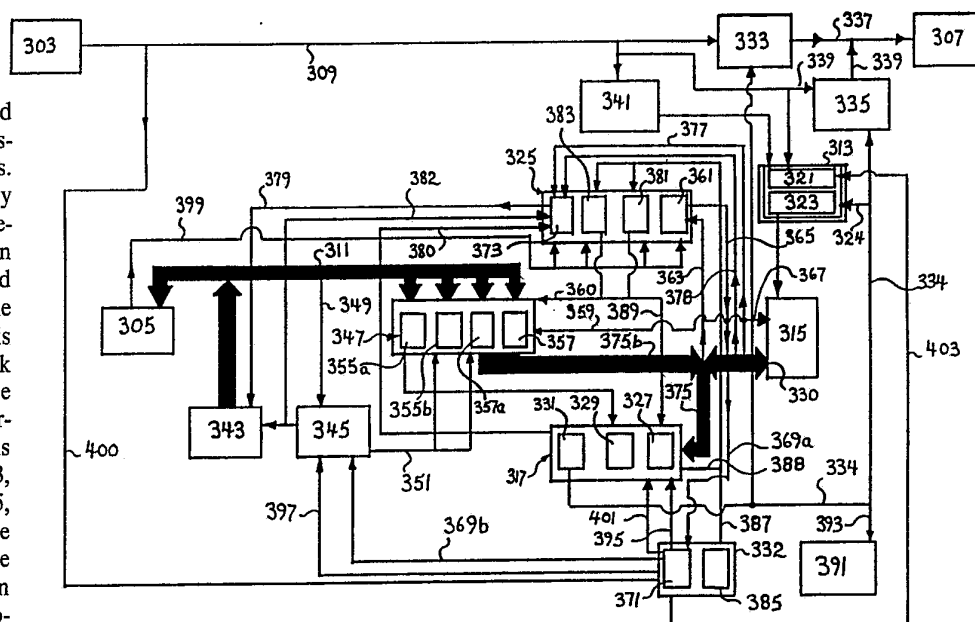
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁵ : G06F 12/14, 12/16, 12/08	A1	(11) International Publication Number: WO 93/02419 (43) International Publication Date: 4 February 1993 (04.02.93)
(21) International Application Number: PCT/AU92/00360 (22) International Filing Date: 16 July 1992 (16.07.92) (30) Priority data: PK 7247 16 July 1991 (16.07.91) AU PL 2927 12 June 1992 (12.06.92) AU (71) Applicant (for all designated States except US): J.A.S. TECHNOLOGY (AUSTRALIA) PTY. LTD. [AU/AU]; 202 Hampden Road, Nedlands, W.A. 6009 (AU). (72) Inventor; and (75) Inventor/Applicant (for US only) : WILSON, Craig, Stuart [AU/AU]; 11 Lucerne Gardens, Edgewater, W.A. 6027 (AU). (74) Agents: KROUZECKY, Stephen, George et al.; Wray & Associates, 239 Adelaide Terrace, Perth, W.A. 6000 (AU).		(81) Designated States: AT, AU, BB, BG, BR, CA, CH, CS, DE, DK, ES, FI, GB, HU, JP, KP, KR, LK, LU, MG, MN, MW, NL, NO, PL, RO, RU, SD, SE, US, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IT, LU, MC, NL, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, SN, TD, TG). Published <i>With international search report.</i>

(54) Title: PROTECTION SYSTEM FOR COMPUTERS**(57) Abstract**

A system and method for protecting computer systems from computer viruses. The system (301) generally consists of a protection device (17) interposed between the computer and the hard disk drive (15, 307). The protection device (17) is connected between the disk controller (13, 303) and the disk drive (15, 307) to intercept relevant control signals issued by the controller (13, 303) to the disk drive (15, 307) and selectively override the signals in accordance with a prescribed protection scheme to protect unauthorised reading or writing of predefined protected areas of the disk drive (15, 307).

The protection device (17) can also be connected to the main communication bus (305) of the computer to enable the CPU of the computer to drive the initialisation means to configure the protection scheme for the disk. Alternatively, the initialisation means may be implemented in software stored within a protected area of the disk, which software is run early in the power up sequence of the computer system performed from a cold boot. In either case, the initialisation means is operated early in the power up sequence before the disk operating system stored on the disk (15, 307) is loaded into the main memory of the computer to effect normal operation of the computer system.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FI	Finland	ML	Mali
AU	Australia	FR	France	MN	Mongolia
BB	Barbados	GA	Gabon	MR	Mauritania
BE	Belgium	GB	United Kingdom	MW	Malawi
BF	Burkina Faso	GN	Guinea	NL	Netherlands
BG	Bulgaria	GR	Greece	NO	Norway
BJ	Benin	HU	Hungary	PL	Poland
BR	Brazil	IE	Ireland	RO	Romania
CA	Canada	IT	Italy	RU	Russian Federation
CF	Central African Republic	JP	Japan	SD	Sudan
CG	Congo	KP	Democratic People's Republic of Korea	SE	Sweden
CH	Switzerland	KR	Republic of Korea	SN	Senegal
CI	Côte d'Ivoire	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
DE	Germany	MC	Monaco	US	United States of America
DK	Denmark	MG	Madagascar		
ES	Spain				

PROTECTION SYSTEM FOR COMPUTERS

This invention relates to computers, and has particular although not exclusive application in the field of personal computers.

In this patent specification, a computer system is defined to include a computer, having a central processing unit (CPU) and read write store. A computer is deemed to include microcomputers, minicomputers, dedicated computers and large main-frame computers.

A "cold boot" is defined to be the action of coming from a power off to a power on stage or equivalent thereof.

Read write is defined to be the act of transferring information from (read) or to (write) a memory or I/O subsystem to (read) or from (write) a central processor unit (CPU).

Fundamental to the operation of a computer of the type to which the subject invention is applicable, is a read write store or memory. The read write store is characterised in being provided with a plurality of data storage locations which may be specified by one or more coordinates. Included in the computer is a means to address the data storage locations, so that they may be selectively read in order to obtain stored information, or have information written thereto.

The read write store is generally divided into two principal types: main memory and mass memory. Main memory is used to store essential system information required for the functioning of the computer, and for storing computer programs currently being executed and the information required for or arising from the running of these computer programs.

The system information forms the basic instructions which define the direction of the CPU's processing flow. Protocol details are also stored within the main memory to give information relating to correct communication with the mass memory, enabling access thereto.

The mass memory is used to store computer programs and information, or portions of them, that are not immediately required by the CPU or simply cannot be accommodated within the main memory. Ideally, a program or information is stored in the mass memory for permanent storage and loaded into the main memory for execution.

The personal computer developed by IBM enjoys considerable popularity, units being owned by people in many countries, and in all walks of life. The personal computer has come to be known as the IBM PC, and there are many IBM PC compatible computers (clones) produced by many companies, the original IBM PC and the compatibles all observing a common protocol in order that there is programming and operational compatibility.

In the IBM PC and IBM PC compatibles, the mass memory is often a hard disk device. The hard disk is connected to a controller therefor, technically defined as a device interface, which is adapted to address data storage locations therein. The data storage locations can be defined as being accessible through a set of three coordinates, one coordinate being directed towards the head number, another coordinate being directed towards the track number or cylinder number, and the final coordinate being directed towards the sector number.

In recent years, there has been a proliferation of self replicating, corrupting computer programs, commonly known as viruses. These viruses can "infect" a computer system,

being loaded into the computer during, for example, downloading of computer programs into the computer. Whilst these viruses have predominantly manifested themselves primarily in IBM PCs and compatibles, they are now also affecting other types of computer systems.

Whilst some viruses may be merely annoying but otherwise harmless, there are others that are more sinister. Viruses within this latter category are those which instantly cause data corruption within the contents of files, those which are delayed and will cause perhaps unrecoverable damage to data after a period of time, and those which are deceptive in character, simulating hardware malfunctions while also causing loss of data.

There are virus protection products available which can detect and remove viruses from computer systems. The success of these known virus protection products depends on the operating environment, which in the case of the IBM PC and compatibles is mostly DOS, being absolutely free from virus contamination. These known virus detection products are useful insofar as that they can detect and remove known viruses, but due to the way the IBM PCs and compatibles have their start-up procedure sequenced, at a cold boot or power-up, all software based protection products must wait for the start-up sequence to be completed, before they can be called upon creating a period of time during which there is no protection against virus contamination.

There are also plug in option cards which use ROMs (Read Only Memory) which may be called earlier in the power-up sequence than purely software based products. The service provided by these cards are equal to other software products in that the code is stored on a different medium i.e. ROM. The links, however, into this ROM originate in

RAM (random access memory) and therefore still suffer the same shortcomings as other software.

Viruses can and do infect a computer system during the start-up procedure, thus infecting the operating environment. Running software based virus detection products in an infected operating environment may result in the detection software itself becoming infected causing either malfunctions in the detection software, or causing it in turn to infect the very software that it is supposed to be checking.

Accordingly, it is an object of the present invention to provide a degree of protection to a computer system from certain types of unauthorised reading or writing from or to a read write store thereof.

It is a preferred object of the present invention to provide a clean environment for at least part of the operating system of a computer.

It is a further preferred object of the invention to provide protection for select parts of the read write store of a computer.

In accordance with a first aspect of the present invention there is provided a protection device for a computer communicating with a read write store for protecting a predetermined portion of said read write store, said protection device comprising: input means to receive address information from said computer; means for determining when said computer is addressing said predetermined portion from said address information; and means for disabling said read write store from being written to, or alternatively from being read from, depending upon the type of protection being provided, when

said means for determining determines that said computer is addressing said predetermined portion.

Preferably said means for determining when said computer is addressing said predetermined portion includes decoder means for providing a flag when said address information corresponds to said predetermined location and said means for disabling includes control means being activated to prevent said read write store from being written to, or alternatively from being read from, as appropriate, in response to said flag when said predetermined portion is being addressed.

Preferably, said input means is connected to the control lines between the computer and said read write store to monitor address information being output by the computer to said read write store.

Preferably, said control means is interposed between the computer and said read write store to intercept selected address information output by the computer and selectively override said selected address information to disable said read write store from being written to, or alternatively from read from, as appropriate.

In accordance with a second aspect of the invention, there is provided a protection system to prevent corruption of computer software instructions relating to the power-up sequence instructions of a computer associated with a read write store comprising:-

(i) a protection device for protecting a predetermined portion of said read write store, said protection device connected between said read write store and said computer and comprising: input means to receive address information from said computer; means for determining when said

computer is addressing said predetermined portion from said address information; and means for disabling said read write store from being written to, or alternatively from being read from, depending upon the type of protection being provided, when said means for determining determines that said computer is addressing said predetermined portion; and

(ii) a storage location provided within the predetermined portion of said read write store for storing said computer software instructions;

wherein said protection device is adapted to prevent said computer software instructions from being altered by preventing said predetermined portion from being written to.

Preferably said computer software instructions include system configuration information.

Preferably said system configuration information includes a partition table.

Preferably said predetermined portion includes further storage locations for storing system operation information.

Preferably said system operation information includes disk operation system information.

Preferably said disk operation system information includes boot sector information.

Preferably said system includes a copy of system operation information stored within said predetermined portion, copied from an unprotected portion of said read write store when said system is installed in said computer, and means

- 7 -

whereby upon power-up or cold boot of said computer a copy of said copy of system operation information within said predetermined portion is copied to said unprotected portion for subsequent use in the power-up sequence, thereby ensuring said system operation information is uncorrupted at least at power-up or at a cold boot of said computer.

Preferably said predetermined portion is adapted to contain a cyclic redundancy check number for each of one or more computer files contained in other unprotected portions of said read write store, whereupon power-up or cold boot of said computer said means is adapted to calculate a fresh cyclic redundancy check number for each of said one or more computer files and compare said fresh cyclic redundancy check numbers with respective said cyclic redundancy check numbers to provide an indication in the event of any one of said one or more computer files being corrupted.

Preferably said indication includes information identifying the computer file which has been corrupted.

Preferably said one or more computer files includes hidden system files and a command interpreter file, said computer being an IBM PC or compatible thereof running DOS software.

Preferably, said read write store is a magnetic disk and said predetermined portion comprises a continuous section of storage area on said magnetic disk of at least one track of one head.

In accordance with a third aspect of the present invention there is provided a method for protecting a predetermined portion of a read write store associated with a computer, said method comprising: monitoring address information of said read write store from the computer; determining when said computer is addressing said predetermined portion from

- 8 -

said address information; and disabling said read write store from being written, or alternatively from being read from, depending upon the type of protection being provided, when determining that said computer is addressing said predetermined portion.

Preferably the determining of when said computer is addressing said predetermined portion includes decoding the address information and providing a flag when said address information corresponds to said predetermined location.

Preferably, the disabling of the read write store is effected by selectively overriding control of the operation of said read write store in response to said flag by causing said predetermined portion only to be read when said predetermined portion is being addressed.

In accordance with a fourth aspect of the present invention, there is provided a method for preventing computer software instructions relating to the power-up sequence of a computer from being corrupted by software corrupting instructions, said method comprising:-

providing a read only storage zone in a predetermined portion of a read write store for containing said computer software instructions;

protecting said predetermined portion of said read write store by: (i) monitoring address information of said read write store from the computer; (ii) determining when said computer is addressing said predetermined portion from said address information; (iii) and disabling said read write store from being written to, or alternatively from being read from, depending upon the type of protection being

- 9 -

provided, when determining that said computer is addressing said predetermined portion; and

copying at least some of said computer software instructions into another portion of said read write store upon power-up or cold boot of said computer.

Preferably said method comprises the step of verifying the status of one or more computer files contained in the unprotected part of said read write store, comprising calculating a fresh cyclic redundancy check number for each of said one or more computer files and comparing said fresh cyclic redundancy check number against a cyclic redundancy check number previously determined for each of said one or more computer files and stored in said predetermined portion, in order to determine if any one of said one or more computer files has been corrupted by corrupting-software instructions.

Preferably said method includes displaying details of any computer files where a variation is found between said cyclic redundancy check number and said fresh cyclic redundancy number, and/or instigating appropriate action directed by the user.

In accordance with a fifth aspect of the present invention, there is provided a method for preventing corruption of computer software relating to and associated with the power-up sequence of a computer communicating with a read write store comprising:

storing code in a storage location of said read write store, which location is accessed by said computer during the initial power-up sequence of said computer to invoke a prescribed subroutine initially upon every power-up or cold boot of said computer;

- 10 -

allocating a predetermined portion of said read write store, including said storage location for protection;

storing a copy of fixed computer software instructions and information essential to the operation of said computer and which is also accessed by said computer during the initial power-up sequence, but which is located outside of said predetermined portion, within said predetermined portion;

storing said subroutine within said protection portion for invoking upon every power-up or cold boot of said computer;

performing a reference check on the validity of prescribed system files containing essential instructions and information for the operation of the computer and read write store; and

storing relevant reference information related to said check within said prescribed portion;

wherein said subroutine causes: (i) a copy of said fixed computer software instructions and information to be written over the existing said fixed computer software instructions and information located outside of said predetermined portion; (ii) a rechecking of the validity of said prescribed system files by performing a fresh reference check and comparing the relevant information obtained therefrom with the original said reference information stored within said prescribed portion; and (iii) performing contingency action if said prescribed system files are found to be invalid.

Preferably said method includes checking the presence of a protection device of the type defined in the first aspect

- 11 -

of the invention connected to said computer and read write store before relinquishing control of said computer.

Preferably said storage location contains the partition table for said computer system.

Preferably said fixed computer software instructions and information comprises boot sector information.

Preferably said prescribed system files comprise hidden disk operating system files and a command interpreter file.

Preferably, said reference check is a cyclic redundancy check and said relevant reference information is a cyclic redundancy check number.

In accordance with a sixth aspect of the present invention, there is provided a protection system for treating a computer system to provide a clean environment for at least part of the operating system of a computer and to thereafter provide protection for select parts of the read write store, comprising:-

a protection device as defined in the first aspect of the invention; and

an initialisation means for operating the computer to perform the method defined in the fifth aspect of the invention;

wherein said initialisation means is activated initially to set up the computer system with a clean environment for at least part of the operating system thereof, before activation of said protection device.

- 12 -

In accordance with a seventh aspect of the present invention, there is provided a protection system comprising:

a protection device as defined in the first aspect of the invention; and

an initialisation means to initially configure said protection device in accordance with a prescribed protection scheme for said read write store.

Preferably, said input means is also connected to the main communication bus of the CPU of the computer to allow for implementing a protection scheme which selectively protects different predetermined portions of said read write store.

Preferably, said means for determining includes a configuration memory comprising a look up table having a series of memory locations corresponding to all of the defined storage locations of said read write store, said memory locations including status information defining said protection scheme for said read write store, an address generator connected to said input means for generating a pointer to said configuration memory, said pointer corresponding to the address of the memory location of said configuration memory which corresponds to the particular storage location of said read write store being accessed by the computer at any prescribed time, and controller means for receiving the status information of the memory location being pointed to by said pointer and analysing said status information to determine whether the computer is addressing a said predetermined portion of said read write store.

Preferably, said means for disabling includes a read/write switch and a step switch, whereby said read/write switch is adapted to prevent said read write store from being written

to, or alternatively read from as appropriate, and said step switch is adapted to prevent a change in the storage location being accessed by the computer from being made, in accordance with the analysis of said status information performed by said controller means.

Preferably, said initialisation means includes a further memory connected to that part of said input means which is connected to the main communication bus of the CPU, said further memory containing executable code to operate said initialisation means, a decoder for controlling communication with said further memory, a communication port connected between said further memory and said configuration memory to buffer communications therebetween, and latching means associated with said decoder, said communication port and said configuration memory to control the sequence of operation of said initialisation means and protection device such that said initialisation means is operated initially to configure said configuration memory in accordance with said protection scheme, and then the protection device is operated to provide ongoing protection during normal operation of the computer system in accordance with said protection scheme.

Preferably, said latching means enacts a switch over between the operation of said initialisation means and said protection device to lock out further communications with said configuration memory and said communication port via the main communication bus of the CPU upon completion of the operation of said initialisation means.

Preferably, said latching means pages between separate halves of said further memory in response to consecutive read accesses of said further memory by the computer after said switch over to encode information stored in either half of said further memory.

Preferably, the protection system includes manual controls to manually override certain status information of certain memory locations of said configuration memory if desired.

Preferably, the protection system includes a display and audio output for providing a visual and/or audio output indicating the status of said protection device during operation thereof.

In accordance with an eighth aspect of the invention, there is provided a method for protecting a computer system comprising a method for protecting a computer system comprising a computer and a read write store, said read write store being configured so that information stored thereon is divided into a plurality of different zones having a prescribed status, each of the zones being further divided into a plurality of different sectors in which the information is stored, said method comprising continuously controlling the accessing of storage locations of said read write store by the computer during the operation thereof, without interfering with the normal operation of the computer, depending upon the type of information stored within a sector and the type and status of the zone within which the sector is located as determined by the configuration of the read write store; wherein said controlling includes allowing, restricting or preventing reading or writing operations by the computer from or to the read write store.

Preferably, the method includes continuously accessing discrete memory locations of a configuration memory defining the configuration of storage locations of the read write store, simultaneously with the accessing of these storage locations by the computer of the computer system; and analysing the configuration definition for the accessed

storage location as accessed within the configuration memory to determine the controlling to be performed.

Preferably, the method includes obtaining control of the computer system during its power up sequence performed pursuant to a cold boot; and configuring the configuration memory for subsequent access.

Preferably, the configuration memory has a series of memory locations corresponding to the defined storage locations of the read write store of the computer system, within which the definition of a protection scheme for said storage locations is stored.

Preferably, the method includes locking the configuration memory from further configuration by the computer after completing the initial configuration thereof, such that only read accesses to the configuration memory are allowed during the normal operation of the computer system.

The invention will be better understood with reference to the following description of several specific embodiments thereof in which:-

Figure 1 is a schematic block diagram of a protection device according to the first embodiment;

Figure 2 is a circuit diagram of a protection device for an IBM PC or compatible computer according to the first embodiment;

Figure 3 is a schematic circuit diagram of the protection device of Figure 2 connected to an IBM PC or compatible computer;

Figure 4 is a flow chart showing the subroutine of the computer software which is run by the computer during power up or cold booting after treatment of the computer system by the software routine shown at

- 16 -

figures 5 to 7 also in accordance with the first embodiment;

Figures 5 to 7 inclusive illustrate parts of a flow chart for the computer software routine, which is run on the computer system to configure it in a manner which facilitates subsequent operation of the protection device of Figure 2 in order to prevent computer software instructions relating to the power-up sequence of an IBM PC or compatible computer from being corrupted by corrupting software instructions;

Figure 8 is a block diagram of an overview of the protection system in accordance with the second embodiment;

Figure 9 is a more detailed block diagram indicating the circuit implementation of the second embodiment;

Figure 10 is a schematic circuit diagram showing the actual implementation of the key processing components and connections of the protection device and initialisation means of the second embodiment;

Figure 11 is another schematic circuit diagram of the actual implementation of the latching means and associated logic circuitry of the memory decoder of the second embodiment;

Figure 12 is a schematic circuit diagram showing the actual implementation of the buttons and the display and audio output of the second embodiment;

Figure 13 is a schematic circuit diagram of the actual implementation of the connections from the controller to the disk drive of the second embodiment;

Figure 14 is a schematic circuit diagram of the actual implementation of the connections to the main bus interface of the computer of the second embodiment;

Figure 15 is a schematic circuit diagram of the biasing circuitry associated with the buttons shown in figure 12 of the drawings;

Figure 16 is a schematic circuit diagram of the

- 17 -

oscillator circuitry used for flashing a LED shown in figure 12 of the drawings;

Figure 17 is a schematic circuit diagram of the duration extension timing circuit for producing the audio input for the display and audio output shown in figure 12 of the drawings;

Figure 18 is a schematic circuit diagram of the actual implementation of the termination circuitry for receiving the WRITE_GATE signal from the controller; and

Figure 19 is a schematic circuit diagram of the actual implementation of the termination circuitry for receiving the STEP pulse signal from the controller.

The embodiment is directed towards a protection system for a computer system incorporating a storage device controller communicating with a read write storage device.

The protection system comprises two discrete aspects. One of the aspects is a protection device which is substantially embodied in hardware form and is semi-permanently incorporated into the hardware make-up of the computer system. The other aspect is an initialisation device and method substantially embodied in computer software for interacting with the computer during a set up phase so as to semi-permanently reconfigure the computer system and alter its power-up sequence instructions in a manner so as to facilitate operation and optimise protection provided by the protection during subsequent operation of the computer system.

The protection device 17, as shown in figure 1, is connected between the read write command signal output 11 from the storage device controller 13 and the read write command signal input 23 of the read write storage device 15 of the computer system. The protection device 17 as

illustrated in figure 3 is configured in a manner applicable to an IBM PC or compatible computer, and not any computer generally.

Referring to figures 1 and 3, the protection device 17 has an input connector 19 connected to the read write command signal output 11 of the controller 13 and an output connector 21 for connected to the read write command signal input 23 of the read write storage device 15. The protection device has input means in the form of a plurality of inputs 25. The inputs 25 are fed to means for determining when the controller is addressing a predetermined portion of the read write storage device, that predetermined portion of the memory containing information to be protected. The means for determining comprises preliminary decoder means 27 and a final decoder means 29. The final decoder means 29 has three inputs 31 for receiving signals from the preliminary decoder means 27. The final decoder means 29 has a control signal output 33 which controls means for disabling the read write storage device 15 from being written to and also for activating a communication circuit 34 to report the protection status of the inputs 25 via audible and/or visual means. The means for disabling is in the form of a control circuit 35 which will override a write command received at the input connector 19 when a control signal is present at the control signal output 33.

Also connected to the final decoder circuit 29 is an enable control means 37 for selectively enabling or disabling the final decoder means 29 to effect overriding control over the protection device. This is provided for enabling information contained in the predetermined portion of the read write storage device 15 to be controlled if necessary.

- 19 -

Each of the three sets of input 25 are decoded by their respective preliminary decoder means 27 to represent a single dimension of the total location address within the read write storage device 15. Each preliminary decoder means 27 is arranged so that the corresponding input 25 is correctly decoded when the particular dimension with which the preliminary decoder means is concerned is addressed by the controller 13. Accordingly, when all of the preliminary decoder means 27 simultaneously decode the particular dimension of the location address of the read write storage device with which they are concerned, the controller 13 is addressing the predetermined portion of the read write storage device which must be protected from being written to, or in the alternative, from being read. When the final decoder means 29 receives decoded signals from all of the preliminary decoder means 27, at its input 31, it then actuates the control circuit 35 to prevent the read write store from inappropriate action depending on the decode information produced by the preliminary decoder means 27, providing that the enable control means 37 has not been selected to override control of the protection device 17.

Referring specifically to the plurality of inputs 25 and the preliminary decoder means 27, each set of inputs 25a, 25b and 25c is intended to merely intercept the address codes for each dimension of read write storage device address information. Each preliminary decoder means 27a, 27b and 27c is involved with decoding address code information input to respective inputs 25a, 25b and 25c.

The operation of the plurality of inputs 25 and the preliminary decoder means will be described in relation to a Winchester type hard disk drive. A single Winchester hard disk drive comprises a number of magnetic disks commonly known as platters, each disk having two opposing

sides which may be exclusively selected for reading from or writing to a head. Each side of each magnetic disk is divided up into a plurality of tracks. The tracks are in turn divided up into a plurality of sectors. Accordingly, the Winchester hard disk drive may be considered to be a read write storage device which has address coordinates for storage locations which may be specified by a three dimensional address location, one dimension relates to the head address, another dimension relates to the track address, and the final dimension relates to the sector address.

The inputs 25a and preliminary decoder means 27a are connected to the complementary binary output of the hard disk controller in an IBM PC or compatible computer, in order that the preliminary decoder means 27a is cognisant of the head selected in the hard disk (read write storage device) which is being selected by the controller. In one embodiment, the preliminary decoder 27a compares the binary information obtained from the hard disk controller, with stored binary information corresponding to the particular hard disk head location which is to be protected. In the present embodiment, the preliminary decoder means 27a merely calculates through logic operations, when a head location to be protected is being addressed.

The inputs 25b and the preliminary decoder circuit 27b are connected to appropriate outputs of the hard disk controller 13 to decode the sector being selected in the hard disk 15. Accordingly, when a sector to be protected is recognised by the preliminary decoder means 27b, a signal is provided to the input 31b of the final decoder means 29.

The inputs 25c and the preliminary decoder means 27c, intercept monitor outputs of the controller 13 which

- 21 -

control the track being selected. When the preliminary decoder means 27c recognises that a track is being selected which must be protected, the output thereof connected to input 31c is provided with a signal for input to the final decoder means 29.

Accordingly, when a predetermined portion of the hard disk is addressed by the controller 13, the input 31 will be signalled by the preliminary decoder means 27, resulting in the final decoder means 29 providing a control signal output to the control circuit 35, which overrides the read write control to the hard disk 15, if not disallowed by the enable control means 37.

In the Winchester hard disk environment, the read write command signal input is a two state signal. One state corresponding to a write command, and the other state corresponding to a read command. Therefore, in the Winchester hard disk drive in an IBM PC, the control circuit 35 will operate to force the hard disk to a read only state to write protect the predetermined portion of the hard disk. Alternatively, if it is desired to read protect the predetermined portion, the control circuit 35 will operate to force the hard disk to a write only state.

Referring now to figure 2, a specific application for an IBM PC or compatible computer will now be described.

In IBM PCs and their compatibles, the first sector of the first track associated with the first head contains the partition table, which comprises information relating to the configuration of the hard disk, which is essential to the operation of the computer. Typically, the other sectors in the first track are unused, however this may not always be the case. If the partition table is corrupted, then either the computer will be misled regarding the

- 22 -

configuration of the hard disk, or the computer will be unable to boot up correctly.

The circuit shown in figure 2 is intended to protect all sectors of the first track associated with the first head. Accordingly, the circuit will protect all information contained in the partition table and any other information contained on the first track, as accessed by the first head. As a result of the determination to protect all sectors of the first track, it is not necessary to determine which sector is being selected by the controller 13. Accordingly, the circuit elements relating to the preliminary decoder means 27b for sector address decoding, and its input, are not required to be included in the circuit. Additionally, as the controller in the IBM PC or compatible hard disk environment has a discrete connection labelled "TRACK 0" which is held at a logic low level when track 0 is detected by the hard disk drive 15, then the complexity of the preliminary decoder means 27c and its input 25c is further reduced.

Referring to figure 2, four inputs 25a are provided for intercepting the head select binary information from the hard disk controller 13 to the hard disk drive 15. Each of these inputs 25 is connected to an inverting buffer 39. The outputs of the inverting buffer are connected to a 4 input OR gate array which comprises two 2 input OR gates 41, the outputs of which are connected to a further 2 input OR gate 43. Accordingly, when the inputs 25a are at a logic high level, corresponding to a BCD output from the controller 13 of 1111, which is asserted whenever the first head (head 0) is selected, the outputs of the inverting buffers 39 will be at a logic low level, and the output of the 2 input OR gate 43 will therefore be at a logic low level. In all other cases, where any one of the inputs is

- 23 -

at a logic low level, the output of the gate 43 will be high.

The inverting buffers 39, 2 input OR gates 41 and 43 collectively define the preliminary decoder means 27a for determining when the first head (head 0) has been selected by the head select binary output signals of the hard disk controller 13.

As the hard disk drive 15 provides a signal which indicates when the first track is selected, this signal being a low logic level provided on the "TRACK 0" connector between the hard disk controller and the hard disk drive, this obviates the necessity for preliminary decoder means 27c. The input 25c intercepts the "TRACK 0" connection between the hard disk controller 13 and the hard disk 15.

The final decoder circuit 29 comprises a 2 input OR gate 45 and a 2 input NOR gate 47. The "TRACK 0" input 25c is fed to one input of the 2 input OR gate 45. The output 49 of the 2 input OR gate 43 (output 49 of the decoder circuit 27a) is connected to the other input of the 2 input OR gate 45. The output of the 2 input OR gate 45 is connected to one input of the 2 input NOR gate 47. The other input of the 2 input NOR gate 47 is held low by resistor 1k (of value 1 kilo ohms), and connected to the enable control means 37 in the form of a jumper 51.

The output of the 2 input NOR gate 47 is connected to the control circuit 35. The control circuit 35 comprises two 2 input NOR gates 53. The output of the 2 input NOR gate 47 is connected to one input of the two input NOR gate 53a. The output of the two input NOR gate 53a is connected to both inputs of the two input NOR gate 53b. The two input NOR gate 53b is configured as an inverter. The other input of the 2 input NOR gate 53a is connected to the input

- 24 -

connector 19 of the protection device, the input connector 19 being provided for connection to the read write command signal output 11 of the hard disk controller 13. The output of the two input NOR gate 53b is connected to the output connector 21 for connection to the read write command signal input 23 of the hard disk drive 15.

The operation of the circuit will now be described. The operation of the preliminary decoder means 27a comprising the 2 input OR gates 41 and 43 has already been described. The important thing is that when the first head is selected, the output of the 2 input OR gate 43 is at a logic low level. If a head other than the first head (head 0) is selected in the hard disk drive 15 then the output of the 2 input OR gate 43 will be at a logic high level.

The final decoder circuit 29, comprising a 2 input OR gate 45 and 2 input NOR gate 47 as discussed is fed the input 25c which is provided for connection to the "TRACK 0" connection of the hard disk controller 13. The "TRACK 0" signal is at a logic low level when the hard disk controller 13 is selecting the first track of the hard disk 15. Accordingly, the output of the two input OR gate 45 will be at a logic low level if the hard disk controller 13 is selecting both the first head and the first track. If the hard disk controller selects any other head or any other track, the output of the two input OR gate 45 will be at a logic high level. If either input of the two input NOR gate 47 is at a logic high level, the output of the two input NOR gate will be at a logic low level. Accordingly, if both inputs of the two input NOR gate 47 are at a logic low level, the output of the two input NOR gate 47 will be at a logic high level.

The jumper 51 comprises a link which may be physically inserted into a socket in the protection device, or removed

- 25 -

therefrom. With the link removed from the protection device, the input of the two input NOR gate 47 to which the jumper is connected is held at a logic low level by resistor 1k. Insertion of the connection in the jumper 51 will result in the output of the two input NOR gate 47 going to a logic low level.

If the output of the two input NOR gate 47 is at a logic low level, the signal which appears at the output connector 21 will be at the same logic level as the signal provided to the input connector 19. Hence the hard disk controller 13 will be able to effect both a read command or a write command at the hard disk 15. However, with the jumper removed and the hard disk controller 13 selecting the first track and the first head, the output of the two input NOR gate 47 will be at a logic high level, and the output of the two input NOR gate 53b will be at a logic high level, forcing the hard disk 15 to a read only state, allowing the first head only to read information on the first track of the hard disk 15. Hence, the partition table, which is contained in this area of the hard disk under convention, will be uncorruptible, providing the connector for the jumper 51 is not inadvertently left in position.

The circuit described with reference to figure 3, if fitted in the manner described, will provide the minimum protection in order to ensure that the partition table in an IBM PC or compatible personal computer is not corrupted.

As the partition table is contained in the first sector of the first track accessed by the first head in the hard disk, by convention, and the remaining sectors are unused, there is considerable scope for protecting further parts of the hard disk memory, by copying other critical information into the unused sectors of the first track accessible by the first head. Consequently, the initialisation device of

the protection system is used to setup the computer in a manner so as to protect further parts of the hard disk memory, making use of these unused sectors which fall within the predetermined portion of the hard disk which is already protected by the protection device.

The circuit described in figure 3, in providing protection to all sectors of the first track accessible by the first head on the hard disk, protects the partition table, but does not protect that portion of the storage device in which the boot sector information for the computer is contained. In the case of the IBM PC and compatibles, boot sector information is normally stored at the head 1, track 0 storage location, which is outside of the predetermined portion protected by the protection device 17, and hence is open to corruption. As the boot sector information is essential to ensuring a proper start up sequence, it is highly desirable to protect this information as well as the partition table. Accordingly, the initialisation device is designed to install a copy of system operation information in the form of boot sector information, in the unused portion of the area of the hard disk protected by the protection device, upon installation of the protection device to the computer.

In addition, the initialisation device is designed to install a cyclic redundancy check number in respect of each of one or more computer files which contain information which does not change in the normal course of operating the computer. The cyclic redundancy check numbers are those for the hidden system files which are commonly known as IBMDOS.COM and IBMBIO.COM or equivalents, and the command interpreter file which is usually known as COMMAND.COM. These files, as with the boot sector information, are critical to the correct operation of the disk operating system, and similarly are not stored within the

predetermined portion of the disk drive. Accordingly the cyclic redundancy check provides a means of checking the validity of these files at power-up.

The initialisation device makes use of the copy of boot sector information and the cyclic redundancy check numbers by placing code into the partition table, which causes a copy of the boot sector information, previously installed into the protected portion of the hard disk, to be written into its normal operating position within the hard disk and to reperform the cyclic redundancy check on the aforementioned files and compare the results with the stored cyclic redundancy check numbers to verify their validity, whenever the computer is powered up or undergoes a cold boot.

The initialisation device is actually in the form of special computer software which is separate run on the computer system when first installing the protection device.

The computer software comprises three programming routines which when executed in conjunction with the computer, set the computer up in a manner so that the computer runs a program sequence subroutine automatically whenever the computer is powered up or subject to a "cold boot", as represented by the flow chart shown in figure 4.

The first programming routine is described with respect to the flow chart shown in figure 5 which represents a program sequence executed for inter alia verifying the validity of certain computer files. The second programming routine is described with respect to the flow chart shown in figure 6, which represents a subroutine sequence of the program sequence shown in figure 5, for installing the computer software in a computer. The third programming routine is

- 28 -

described with respect to the flow chart shown in figure 7, which represents a subroutine sequence of the program sequence shown in figure 5 for decommissioning the computer software in a computer.

Referring specifically to figure 4, after installation of the protection system which includes connection of the protection device 17 and the successful running and treatment of the computer system with the initialisation device software, power-up, the computer during power-up is given a prompt 101 upon reading the code previously entered into the partition table by the initialisation device. This causes the computer to execute a step 103, comprising copying the copy of the boot sector information from the protected area of the hard disk into the normal operating area of the hard disk, and reperforming the cyclic redundancy checks as previously described. After checking the validity of these files, the computer continues with its normal operation.

Now describing the computer software of the initialisation device in more detail, referring to the main software flow chart shown in figure 5, upon the computer receiving a start prompt 105, the software causes the computer to execute a step 107 which sets the flags and initialises the variables. The software then performs a check 109 in order to determine if the computer has a monochrome card for a monochrome monitor. In the event that a monochrome card is found, the computer executes a step 113 which sets the video mode to a monochrome monitor. If a monochrome card is not found, then the software executes a step 115 which sets the video mode for a colour monitor.

The software then causes the computer to perform a check 117 for the number of hard drives. If the number of hard drives is equal to 0, the software steps the computer to

- 29 -

execute a step 119 which causes an error message to be displayed before the program ends. In the event that the number of drives is not equal to 0, the software steps the computer to execute a step 121 which sets the number of drive bits to the value of drives.

The software then causes the computer to perform a check 123 for whether the parameter relating to the number of drives found in the computer during the power-up sequence is 0; if the parameter is not 0, the computer performs a check 125 to determine whether the parameter is 2. If the parameter is 2 the computer performs a check 127 to see if a hard disk drive number selected by the operator is 2. If this is the case then the computer executes a step 129 which sets the drive bit to second drive. In this manner, the second hard disk drive can be protected by the protection system, as well as the first hard disk drive, however, the operator must manually instruct the initialisation device to protect the second hard disk drive.

If the perimeter is not 0, or if the parameter is not 2, or if the number of drives is not equal to 2, then the software continues, assuming that the number of drives is equal to 1.

Following the previous steps, the computer executes a step 131 causing a message "please wait" to be displayed. The computer then performs a check 133 to determine if the cyclic redundancy numbers have been initialised. If the cyclic redundancy numbers have been initialised, the computer executes a step 135 which calculates fresh cyclic redundancy check numbers for the hidden system files and the command interpreter file, and compares these fresh cyclic redundancy check numbers against cyclic redundancy check numbers previously stored for the hidden system files

- 30 -

and the command interpreter file when the software and protection device was first installed. If there is no variation between cyclic redundancy check numbers, the software causes the computer to set a flag "true", however if there has been a change, the computer sets a flag "false".

If the check 133 reveals that the cyclic redundancy check numbers have not been initialised, the software causes the computer to execute a step 137 which generates cyclic redundancy check numbers for the hidden systems files and the command interpreter file and stores them in an unused portion of the protected portion of the hard disk, protected by the protection device.

The computer then executes a check 139 in order to determine whether the cyclic redundancy check numbers are secure. If not, the computer executes a step 141 which causes an error message to be displayed before the program is brought to an end. If the cyclic redundancy check numbers are secure, the computer then executes a check 143 to determine whether the decommission subroutine "P" has been selected. If the decommission subroutine has not been selected the computer then executes a step 145 which causes a logo to be displayed. Following the displaying of the logo, the computer then executes another step 147 which establishes which hard disk drive is active, and the number of hard disk drives present in the system. It then determines the relevant drive data, type of partitioning, whether or not the system is bootable, and then causes defined partition information to be displayed.

The software then steps the computer to perform a step 149 which checks that the protection device is active, and if so sets a flag. The computer then performs a check 151 which determines whether the flag for the protection device

- 31 -

is active. If the flag is active, the computer executes a step 153 which causes an active message to be displayed. If the flag shows the protection device is not active, the computer executes a step 155 to display an appropriate message.

The software then causes a step 157 to be executed by the computer where the active partition table is found. A step 159 is then executed where a pointer is loaded at the boot sector. Then a step 161 is executed where the boot sector is tested for, before a step 163 sets control flags.

The computer then executes a step 165 which tests for an installation command flag which may have been entered by a user, before executing a check 167 in order to determine if the installation subroutine has been selected and the boot sector is valid. If so, then the installation subroutine "Q" is selected. If the installation subroutine has not been selected, then the computer executes the step 169 of generating a cyclic redundancy check number for the command interpreter file COMMAND.COM. The software then executes a check 171 comparing the cyclic redundancy check number generated at step 169 against the reference generated at steps 137 or 135, and stored in the protected portion of the hard disk. If there is a difference, then the computer executes a step 173 which sets an error flag, and displays a message to the effect that the command interpreter file has been corrupted. If the cyclic redundancy check number is unchanged, the computer executes a step 175 displaying a message that the command interpreter file is secure.

The computer then executes a step 177, to generate a cyclic redundancy check number for the first hidden file which is IBMBIO.SYS or an equivalent, before performing a check 179 of the cyclic redundancy number generated in the step 177, against the reference stored in the protected portion of

- 32 -

the hard disk following step 135 or 137. If there is a variation between the cyclic redundancy number generated at step 177 and the reference cyclic redundancy check number then the computer executes a step 181 which sets an error flag and displays a message to the effect that the cyclic redundancy check number concerned is corrupt. If there is no difference between the generated cyclic redundancy check number and the reference, then the computer executes a step 183 which displays a message that the cyclic redundancy check number concerned is secure.

The computer then executes a step 185 to generate a cyclic redundancy check number for the second hidden file which may be IBMDOS.SYS or an equivalent file, before performing a check 187 to compare the cyclic redundancy check number generated at step 185 with the reference cyclic redundancy check number stored in the protected portion of the hard disk following step 135 or 137. If there is a variation between the generated cyclic redundancy check number and the reference, then the computer executes a step 189 which sets an error flag, and displays a message to the effect that the second hidden file is corrupt. If there is no variation between the generated cyclic redundancy check number and the reference, then the computer executes a step 191 which displays a message to the effect that the second hidden file is secure.

The computer then executes a check 193 to determine if any error flags are set. If error flags are set, the computer executes a step 195 which sets the error level, and displays an appropriate message, before the program finish (end). If the error flags are not set, the computer performs a check to determine whether the protection device is active. If the protection device is not active, then the computer executes a step 199 to set the error level and

display a message before the program finish. If the protection device is active, then the program finishes.

Referring to figure 6 the installation subroutine is shown. If the installation subroutine has been flagged and this has been detected at step 165, and if the boot sector is valid, then the installation subroutine, subroutine Q is selected. The first step of the subroutine is a check 225 that the protection device is active. If the protection device is active, the computer executes a step 227 which causes an error message to be displayed, before the program finishes. If the protection device is not active, the computer executes a check 229 to determine that the DOS level is greater than 2, for reasons discussed later. If the DOS level is greater than 2, the computer executes a check 231 to determine if the boot sector is valid. If the boot sector is valid, the computer executes a step 233 which causes the boot sector information to be copied into the area of the hard disk protected by the protection device. The computer then executes a check 235 in order to determine if the copy is okay, before the program continues. If the check 229 for the DOS level shows that the DOS level is not greater than 2 then the computer executes a step 237 which causes error flags to be set. If the check 231 reveals that the boot sector is not valid then the computer executes the step 237 to set error flags. If the check 235 reveals that the boot sector copy is not okay, then the computer executes the step 237 to set the error flags.

The computer then executes a check to determine if any error flags are set. If the check 239 determines that error flags are set, then the computer executes a step 241 which causes an error message to be displayed, before the program ends. If the check 239 reveals that no error flags are set, then the computer executes a step 243 which

- 34 -

generates a cyclic redundancy check number for the command interpreter file COMMAND.COM. The computer then executes another step 245 to generate a cyclic redundancy check number for the first hidden file, before executing a further step 247 to generate a cyclic redundancy check number for the second hidden file. The computer then executes a step 249 which saves all cyclic redundancy check numbers in an unused portion of the protected area of the hard disk, before executing a step 251 causing a message to be displayed showing that the installation has been successful, before the program ends.

Referring now to figure 7, if the check 143 determines that the decommission subroutine has been selected, then the computer executes a step 201 causing appropriate flags to be set. The computer then executes a check 203 to determine that the DOS version level is greater than 2. If the check 203 determines that the DOS level is not greater than 2, the computer executes a step 205 causing an error message to be displayed before the program ends. If the check 203 determines that the DOS version level is greater than 2, then the computer executes a step 207 to test for the copy of the boot sector information in the protected portion of the hard disk. The computer then executes a check 209 to determine if the boot sector information copy is present. If the check 209 reveals that the boot sector information is not present, the computer executes a step 211 causing an error message to be displayed before the program ends. If the check 209 reveals that the boot sector information copy is present in the protected portion then the computer executes step 213 causing the boot sector information copy to be installed in the normal location for accessing boot sector information on the hard disk. The computer then executes a step 215 causing the backup copy of the boot sector information contained in the protected portion of the hard disk to be erased.

The computer then executes a check 217 to determine whether the erasure of the copy of the boot sector information from the protected portion of the hard disk has been successful. If the check 217 shows that the erasure has been successful, then the computer executes a step 221 causing a message to be displayed showing that the erasure was successful, before the program ends. If the check 217 shows that the erasure was not successful, then the computer executes step 219 which causes an error message to be displayed, before the program ends.

The protection system including the protection device 17 shown in figure 3, and the initialisation device, ensures that an IBM PC, when in use, can not have its operating environment corrupted by a virus or other type of software. In addition, the software itself provides a check, utilising the advantage of having a protected area on the hard disk which is provided by the protection device, in order to determine that the command interpreter file and the hidden system files are secure. Given the foregoing disclosure, it is easy to envisage either expanding the scope of the protection device in order to encompass protecting more areas of the hard disk, however the complexity of the protection device would necessarily be increased due to the necessity of the protection device to have cognisance of the area of the hard disk being addressed, utilising the protocol used by an IBM PC. In a computer system utilising a different protocol, or a different type of read write store, however the further complexities could be obviated. In addition, it is clear that the protection system could be expanded to protect further files by use of the principle of cyclic redundancy number checking as utilised for the command interpreter file and the hidden system files.

The cyclic redundancy check numbers are generated using known algorithms for providing reference numbers for use in file checking. Any security breach by a virus to any of the files for which a cyclic redundancy check number is stored, will result in any fresh cyclic redundancy check number which is calculated showing a variation from the stored cyclic redundancy check number. As a consequence, the software is able to recognise whether any such file has been corrupted by a virus.

It should be noted that the special computer software of the initialisation device can be permanently left in the computer system after set up of the protection system so that in the IBM PC environment, it can be run in a modified form during the power-up sequence, after the subroutine performs the writing of boot sector information copy into the normal position in the hard disk, from the boot sector information stored in the protected portion. In this mode, the special computer software is stored in the computer file AUTOEXEC.BAT which should be run after the aforementioned subroutine sequence. In this file, the computer software as shown in the flow chart in figure 5 is the first file to be run. Once the software shown in figure 5 has been run, then the normal initialisation procedure of the computer can continue.

It will have been noted that the software specifies a DOS version level greater than 2 (actually level 3.0 and greater). This is due to the fact that before DOS version level 3.0, there was no preferred defined format for running hard disks in an IBM PC environment. Accordingly, the position of storage of disk operating system programs was not standardised. Whilst this does not mean that the device, system or methods of this invention could not be adapted to earlier versions of DOS, the specific device shown in figure 3 together with the software forming the

system, is intended to be an off-the-shelf product which can be installed in the compatible computers. MFM (modified frequency modulation) and RLL (run link limited) are two specific types of hard disk drive which are envisaged as being protected under the system.

The jumper 51 is an internal link which is only physically accessible. This is in order to ensure that the protection device can not be overridden by software means, for example such software means as corrupting software which would result in the protection device and the initialisation software being circumvented. Referring to figure 1, the jumper 51 is shown in the form of the enable control means. The form of control exercised over this enable control means may simply be the jumper 51. However any form of control may be used, so long as the control is not under the control of software which may be corrupted by a computer virus.

A limitation with the preceding embodiment is that the predetermined portion of the read write store is specified first and, in an IBM PC environment, is designated to be that portion of the hard disk which includes the partition table. In practice, this portion is actually disposed on the first track of the first head of the hard disk and includes a number of sectors which normally are unused but in the embodiment are utilised to store additional information, code and files which are desired to be protected. Those aspects of the additional information, code and files which would normally reside outside of the predetermined portion, are copied from their normal storage location to the predetermined portion of the hard disk which is to be protected, so that the duplicate copy can be used as a master which is protected during normal operation of the system and which is used to overwrite the normal code (unprotected) during power-up to ensure the

authenticity of the normal code (unprotected) at least at the commencement of normal operation of the computer, after power-up. Consequently, it is necessary to treat the hard disk by the initialisation means, and adapt it to suit the constraints of the protection system, as opposed to adapting the protection system to the constraints of the hard disk.

In order to overcome this limitation, and make the invention more utilitarian, the invention is embodied in a second and more preferred form, which will now be described with reference to figures 8 and 9 inclusive.

The second embodiment is directed towards a protection system for protecting any number of predetermined portions of the read write storage device of a computer whereby the interface in between the storage device controller of the computer and the read write storage device is characterised by the use of an MFM or RLL encoding scheme for standard interfacing schemes or alternatively for the enhanced small device interface (ESDI) interfacing scheme. The embodiment takes two discrete forms, which are marginally different from each other, one form being for standard interfaces for MFM/RLL encoding, such as ST-506/412 interfaces, and the other form being for the ESDI interface.

As shown in figure 8 of the drawings, the protection system 301 is interposed between, on the one hand, the storage device controller in the form of a hard disk controller 303 and the main bus interface 305 of the computer (not shown), and on the other hand, the read write storage device in the form of a hard disk drive 307.

As with the protection system of the preceding embodiment, the protection system 301 comprises a protection device which operates during run time to protect the disk drive,

and an initialisation means which operates at power up from a cold boot to set up and configure the protection scheme for the disk drive. The protection device includes input means to receive address information from the computer, means for determining when the computer is addressing a predetermined portion of the hard disk 307 which is to be protected, and means for disabling the disk drive from being written to, or alternatively being read from, when the means for determining determines that the computer is addressing one of these predetermined portions of the disk drive. The initialisation means is markedly different from that of the preceding embodiment, whereas instead of being implemented within the hardware of the computer system by virtue of special computer software, in the present embodiment it is partly embodied in peripheral hardware forming part of the protection system and makes use of the protection device in conjunction with the computer system to set up and configure the protection scheme for the disk drive.

The input means is somewhat more sophisticated than in the preceding embodiment, whereby address information is obtained not only from the drive interface between the hard disk controller 303 and disk drive 307, but also from the main bus interface 305 in order to accommodate the initialisation means.

The address information obtained from the drive interface is received along main input lines 309 and includes five essential pieces of information which are drawn directly off the drive interface, namely the drive selected, the head selected, the read/write select function, the head direction, and the head stepping pulses, all essential for controlling the operation of the disk drive 307. The actual connections for a standard interface using MFM/RLL encoding from the controller 303 and to the disk drive 307

are shown in figure 13 of the drawings. Moreover, socket J1 provides the connections from the controller 303 and socket J2 provides the connections to the disk drive 307. As can be seen from this diagram, the WRITE_GATE and STEP lines from the controller do not connect directly to the WRITE_GATED and STEP2 lines to the disk drive so that the protection device can impose control over the reading, writing and stepping of the disk drive. Accordingly, appropriate termination of the WRITE_GATE and STEP signals is provided by open collector circuits as shown in figures 18 and 19 of the drawings.

The address information obtained from the main bus interface 305 is communicated to the protection system 301 along the main communication bus 311, directly from the bus interface 305.

The bus interface is a standard communication means comprising an option card having an edge connector which is directly connected into a bus slot of the computer, thereby linking the CPU of the computer directly via its main communication bus to the protection system 301. The need for this connection, contrary to the previous embodiment, is that by these means the protection system can interact with the CPU of the computer to, inter alia, set up the configuration of the protection scheme for the hard disk drive during operation of the initialisation means, and thus enable a plurality of discrete predetermined portions of the memory of the disk drive 307 to be protected. This extends the utility of the protection system considerably than was the case in the preceding embodiment.

In the actual implementation of the present embodiment, the computer is an IBM PC and the main communication bus is the IBM bus. The edge connector of the bus interface 305 connects to a dual in-line socket J10 as shown in figure 14

- 41 -

of the drawings, which identifies the various connections to the protection system outside and on either side of the socket J10 and their corresponding connections to the IBM bus lines on the inside of the socket.

With respect to the relevant connections shown with the sockets J1, J2 and J10, these connect to relevant parts of the actual implementation of the protection system as shown in the various circuit schematics. Accordingly in order to facilitate understanding of the various connections, the same reference names have been used for different connection lines of the circuits to show that these lines are interconnected. For example, as previously described the WRITE_GATE line connection at J1 connects to the WRITE_GATE line connection shown in figure 18. Similarly the STEP line connection shown at J1 connects to the STEP line connection shown in figure 19.

The means for determining when the computer is addressing predetermined portions of the disk drive memory to be protected operates on the address information generated by the hard disk controller 303, and generally comprises an address generator 313, a configuration memory 315 and a zone controller 317. The configuration memory 315 in the actual implementation of the present embodiment comprises a 5256 dynamic RAM chip U11, as shown in figure 10 of the drawings. The address generator 313 and zone controller 317 are actually implemented within a field programmable gate array logic integrator circuit (FPGA) U1 also shown in figure 10 of the drawings. As the FPGA circuit is a hardware programmable circuit the address generator 313 and zone controller 317 will be described specifically by way of their function.

The address generator 313 essentially uses the drive select, head select, head stepping and head direction

information generated by the disk controller 303 and input via input lines 319 which form part of the main input lines 309. It should be appreciated that in the actual implementation of the protection system the input lines 319 do not all directly connect to the FPGA U1, but are connected to other circuit elements which in turn are connected in one way or another to the FPGA U1. Within the FPGA U1, the address generator 313 comprises appropriate logic 321 to drive appropriate up down counters 323 which generate a binary address corresponding to the finite location currently being accessed by the disk drive. This binary address is used as a pointer for a look up table stored with the configuration memory 315 which will be described in more detail later. Operation of the address generator 313 is also controlled by a park signal input via control line 324 from the zone controller 317, the function of which will also be described later.

The configuration memory 315 is programmed in a manner to be described later to contain a look-up table comprising a series of memory locations corresponding to all of the valid storage locations that are to be defined for the disk drive and the status information to be stored in respect of each such location. The status information comprises eight bits of data, five bits devoted to defining user or system area read or write status of the particular location, one bit defining whether stepping up from the location is permitted, one bit defining whether the location is to be used to enable operation of a paging system to be described later, and one bit defining a flag signifying whether the configuration memory is to be locked out from further configuration of the memory locations after the status for that particular memory location has been stored. The relevance of the lock-out flag will be understood later.

The configuration memory 315 is driven by the address generator 313 and also a latching means 325 to control reading of information from the data transfer port 330 of the configuration memory by the zone controller 317 along the internal bus line 375a, or writing of information thereto by the initialisation means along the internal bus line 375b.

The zone controller 317 comprises three sequential processing means, a manual override processor 327, a zone section processor 329 and a zone analysis processor 331. These are formed by suitable logic processor circuits within the FPGA U1, the function of which will presently be described.

The zone controller 317 is essentially designed to receive the status information read from the configuration memory 315 for the particular memory location within the look-up table contained therein, being pointed to by the address generator 313, analyse this information by allowing for any overriding or combining of this information with other information having regard to other input parameters, and produce a set of control signals to drive the means for disabling the disk drive from being written to or read from as a result of the analysis.

The manual override processor 327 and the zone section processor 329 of the zone controller provide additional functions to the means for determining when the computer is addressing predetermined portions of the disk drive memory.

In the case of the manual override processor 327, this is connected to the data transfer port 330 of the configuration memory 315 via the internal bus line 375a to read status information from a memory location pointed to by the address generator 313. In addition, the manual

override processor 327 is connected via input control lines to selected outputs of the latching means 325 and to a set of manual controls 332 which will be described in more detail later.

The function of the manual override processor 327 is to process manual override signals input to the protection system 301 via manual controls 332 to dictate to the protection device exactly what information to process for the memory location being pointed to by the address generator 313, the corresponding storage location for which is currently being addressed by the controller 303. More specifically, the manual override processor 327 controls the combination of non-system protection data stored for the memory location pointed to, by switching relevant protection data for this location in accordance with the particular state of certain latches of the latching means 347 which are set by the manual controls, whilst not reducing the level of protection provided for system defined memory locations which are pointed to. For example, the status information stored within the configuration memory 315 may indicate to always allow a write request from the disk controller 303 through to the disk drive 307 for this particular storage location of the disk drive. However, if the manual controls 332 are selected so as to provide an override signal indicating that all write requests are to be disallowed, then the manual override processor 327 operates to issue a signal to the zone section processor 329 indicating that this particular storage location is to be given a read only condition. Consequently, all memory locations pointed to by the address generator 323 during the issuance of such a control signal via the manual controls 332 will be overridden by the manual override processor 327 to provide for a read only condition.

With respect to the zone section processor 329, it accepts input signals from the output of the manual override processor 327 and also from one of a set of configuration ports 347 which form part of the initialisation means and will be described in more detail later. The function of the zone section processor is essentially to allow for additional information pertaining to the particular zone or partition within which the storage location being accessed by the controller 303 falls, so that this information is combined with the status information stored within the configuration memory 315. For example, this additional information which is provided by the initialisation means during setting up of the protection scheme may indicate whether a particular zone or partition of which the storage location is a member, is to be enabled or disabled. The zone section processor 329 provides the means by which this additional information can be channelled from the particular port of the configuration ports 347 dedicated to holding this information for input to the zone analysis processor 331 in conjunction with the status information output by the manual override processor 327.

The zone analysis processor 331 has its inputs connected to the outputs of the zone section processor 329, and provides the main output from the zone controller 317. The zone analysis processor 331 receives the combined zone or partition data from the zone section processor 329 with the status information for the particular storage location being addressed by the controller 303 and produces a set of controlling signals which are output along control lines 334 to drive the means for disabling the disk drive from being written to or read from in accordance with the status information and zone information input thereto. In addition, the zone analysis processor 331 provides the park signal input to the address generator 313 via control line 324, as previously described, and display and audio signals

- 46 -

for driving a display audio output 391 via control line 393, the function of which will also be described in greater detail later.

The means for disabling the read write store from being written to or read from essentially comprises a read/write switch 333 and a step switch 335. The read/write switch 333 and step switch 335 are actually implemented within a programmable array logic circuit (PAL) U2 as shown in figure 10 of the drawings, which operates more quickly than the FPGA U1. These switches are both controlled by the zone analysis processor 331 and are respectively connected in line with the read write select line 337 (WRITE_GATE-WRITE_GATED) and the head stepping line 339 (STEP-STEP 2) which form part of the main lines 309 of the drive interface interconnecting the hard disk controller 303 and the disk drive 307. These switches comprise logic gates which either allow the corresponding signal output by the hard disk controller 303 through, on the respective read write select line 337 or head stepping line 339, or alternatively override the signal as determined by the zone controller pursuant to the analysis performed thereby as previously described.

Although the purpose of the read/write switch 333 is self evident, the purpose of the step switch 335 requires further explanation. Moreover, the step switch 335 prevents the particular head of the hard disk 307 being accessed by the controller 303, from being made to step further past its last track (park track) on the hard disk, even when further head stepping signals are issued by the controller 303. This is achieved as a consequence of the protection device being cognisant of the limitations of the disk drive 307 by virtue of the configuration scheme implemented by the initialisation means, where the configuration memory 315 is provided with status

information for the park track locations of the disk drive 307 during initialisation. Consequently, when the head of the disk drive currently being accessed reaches the park track of the disk, the step switch overrides the disk controller 303 suppressing any further head stepping signals issued by the controller by receipt of a park track control signal output by the zone controller 317 as a result of it analysing the status bit information indicating whether the head can step up from the track or not. As previously mentioned, this status information is stored within the memory location of the look-up table of the configuration memory 315 corresponding to the disk storage location being accessed by the head controller and pointed to by the address generator 313.

In addition to the step switch 335 receiving the park track control signal, as previously described, the address generator 313 receives the park track signal to prevent further upcounting of the counters 323 thereof. Consequently, the address generator 313 is locked from upward counting for the duration that the step switch 339 suppresses any further head stepping signals which would try and move the head beyond the park track, and in this manner, the address generator is prevented from getting out of synchronisation with the actual physical tracking of the head on the disk drive 307. Importantly, if the address generator was not disabled from further upcounting of incoming head stepping signals, the address generator 313 would continue to count head stepping signals, even though the head could not physically move beyond the park track of the hard disk, and would bring the pointer out of synchronisation with the physical location of the head, hence making the protection scheme useless and in fact destructive to the continued operation of the computer system.

It should be noted that in the case of the standard interface used for MFM/RLL encoding, the input means requires no sophistication as the relevant information that needs to be decoded for determining and tracking the specific memory location being accessed by the controller 303 at any one time is provided directly by the interface lines 309. In the ESDI interface, however, (not shown in the schematic circuit diagrams of the actual implementation of the protection system) this is not the case and accordingly it is necessary to incorporate some logic into the protection system 301 in order to correctly decode the different information that is provided on the drive interface lines 309 in the ESDI scheme. Moreover, a processor 341 is interposed between the drive input lines 309 and the address generator 313 to receive the relevant control signals output from the controller 303 and decode them in a manner so as to extract the drive select, head select, read/write, step, and direction information which is subsequently output from the processor 341 for input to the address generator 313. In practice, the drive select, head select and read write select information is substantially the same as in the standard interfacing scheme using MFM/RLL encoding, however the stepping and direction information is not provided and needs to be reconstituted from other lines provided in the ESDI interface. Apart from this difference, the protection system described in the present embodiment is the same for standard interfacing using MFM/RLL encoding and ESDI interfacing.

The initialisation means, which instead of being of software form as was the case in the preceding embodiment, is principally embodied in hardware form and makes use of the CPU of the computer (not shown) by means of the bus interface 305.

As shown in figures 8 and 9, the initialisation means generally comprises the bus interface 305, a device ROM 343, a memory decoder circuit 345, and the configuration ports 347 which are controlled via the latching means 325 to work in conjunction with the remainder of the protection system 301. The essential purpose of the initialisation means is to initially configure the memory locations contained within the look-up table of the configuration memory 315 in accordance with the protection scheme employed for protecting the defined storage locations of the disk drive 307 during the normal operation of the computer system, to which these memory locations correspond. The initialisation means is operated during the initial power-up sequence of the computer system, and is designed so that after it has completed the configuration of the configuration memory 315 in accordance with the protection scheme, it locks out intelligible communication between the computer and the protection system via the bus interface 305.

The device ROM 343 in the actual implementation of the present embodiment is in the form of a 27256 EPROM U9 and is connected to the communication lines 311 of the IBM bus so that information may be read therefrom by the computer via the bus interface 305. Alternatively, the computer may write directly to the configuration ports 347 via the bus interface 305 and the main communication bus 311. The code stored within the memory of the device ROM 343 contains signature recognition information and ROM usage data which is accessed by the computer during power-up to enable the device ROM to gain control of the power-up sequence by emulating the configuration of the basic input output system (BIOS) extension ROM. In this manner, the protection system and in particular the initialisation means is operated very early in the power-up sequence when the CPU

- 50 -

of the computer tests for valid device ROMs, and if found, passes control over to them.

In this respect, the signature recognition information of the device ROM 343 is searched for during the power-up sequence of the computer to enable additional hardware to the computer system such as the protection system to convey to the computer how to communicate to this additional hardware, which otherwise is unknown to the computer. During the start-up sequence of the computer, this search for signature recognition information contained within the device ROM 343 occurs at 2K byte boundary steps of the overall memory map of the computer system, therefore making the minimum size of the device ROM 2K bytes.

The device ROM 343 is programmed so that once the code and the device ROM has been executed, the CPU of the computer is instructed to load information from the hard disk drive, this information then being used for directing the flow of the device ROM code. Accordingly, a number of useful sub-routines adapted for enhancing the protection and security of the computer system can be run by means of the device ROM 343 at this early stage of power-up, which need not all be stored on the device ROM. Sub-routines which can be run as executable code, stored either on the device ROM or the disk drive as accessed by the device ROM, include the following:

- (a) CMOS validity test and correction program
- (b) Password filter program
- (c) Virtual environment switching program
- (d) Loading of executable code for configuring the configuration memory 315 in accordance with the protection scheme
- (e) Natural device ROM termination program

The execution flow of these programs will be described in more detail later.

The memory decoder 345 is also implemented within the FPGA U1, and comprises logic circuitry, the principal function of which is support circuitry for the device ROM 343 and also for the latching means 325. The memory decoder 345 controls communications to the configuration ports 347 via output line 351 and from the device ROM 343 via output line 353, whilst monitoring address line information from the CPU via input line 349 which is connected to the bus line 311. In addition, the memory decoder 345 receives a control signal from the latching means 325 via control input lines 369a and 369b, to effect lock-out of the configuration memory 315 from communication with the configuration ports 347 during switch over, a process that will be described in more detail later.

The memory decoder 345 continually monitors the CPU's address signals via the bus line 311 to determine whether there is a match with a predefined addressing range. If a match is detected, the memory decoder further analyses the data on the bus line 311 to determine whether the access is a read or write attempt. If the access is a read request, the memory decoder is designed to enable the device ROM to be read. Alternatively, with a write request, the device ROM is designed to enable the configuration ports 347 to be written to.

The configuration ports 347 are divided into two principal ports, one 355 dedicated to receiving even address information, and the other 357 for receiving odd address information. In the actual implementation of the present embodiment, the two ports are in the form of respective registers U8 and U7 as shown in figure 10. The even port 355 (U8) is further divided into two halves or buffers, one

- 52 -

buffer 355a (locations 0 to 3) being adapted to receive zone definition information, and the other 355b (locations 4 to 7) being empty. The odd port 357 (U7) is not divided into two physical halves or buffers, but comprises a single buffer (locations 0 to 7) adapted to receive two different types of buffered information, one type of buffered information 357a being configuration definition information and the other type of buffered information 357b being control switch information and system protection information. The registers/buffers are tri-state and are controlled by appropriate decoder circuitry U16B, U17A, U16C, U16D, U4A as shown in figure 11, associated with the memory decoder 345, via output lines 351 (CONFIG, PORT) and the latching means 325 via output 359 (BUF) and 360 (UPDAT). Information is transferred in the one direction only from the bus 311 through to either the configuration memory 315, the latching means 325 or the zone controller 327, depending upon the nature of the information being input to the configuration ports 347 and the sequence of operation of the initialisation means. Thus, the configuration ports 347 are a write only circuit which complement the read only of the device ROM 343, and provide a window through which all direct communications between the computer system including the device ROM 343, and the protection device are conducted.

As previously stated, the ports are divided into an even port 355 (U8) and an odd port 357 (U7) by mapping the even and odd addresses of the protection system within the window defined by the device ROM 343. Thus, even address information is directed towards the even port 355 and is stored within the zone definition buffer 355a, this information defining how many zones, i.e. system and user areas, are to be selected for the computer system and which of these are to be protected. This is primarily of relevance for the display and audio output 391 and the

update function that can be initiated via the manual controls 332, which will be described in more detail later. Each data bit comprising the zone definition information is assigned to a zone, and the enabling or disabling of write protection to the zone is signified by the digital value of this bit.

Odd address information is directed towards the odd port 357 and is split up to be stored into the configuration definition buffer 357a and the control switch buffer 357b. The configuration definition information and information from the control switch buffer 357b is intended to be input to the configuration memory 315 for storage therein, in accordance with the sequence of operation performed by the initialisation means when setting up the protection system to protect the disk drive 307, and the control switch information is used to trigger a switch over of the protection system at the completion of the initialisation sequence performed by the initialisation means and thereafter lock out the configuration ports 347 from communicating any further information from the bus 311 to the protection device, and in particular to the configuration memory 315, latching means 325 and/or zone controller 327.

The actual implementation of the latching means 325 is shown in figure 11 and comprises four discrete latches U10B, U10A, U6B and U6A each of which are implemented in the form of negatively edged JK flip flops as shown in figure 11 of the drawings. The latching means 325 provides the means of controlling the operation of the protection system in accordance with the sequence of operation of the system, which is programmed into the executable code stored within the device ROM 343 or within the disk drive 307 and accessed by the device ROM in the manner previously described.

The first latch is a mode latch 361 (U10B) which is used to lock out further changes to the configuration memory 315 by receiving the triggering control signal output by the control switch buffer 357b (U7) along the internal data bus 375b. The JK flip flop U10B for the mode latch 361 is specially configured in a type of one shot arrangement so that once the flip flop is triggered, it will maintain its set outputs until reset via the reset line 399 (PWRRESET). The triggering control signal actually comprises a data bit which is used as a flag, this data bit firstly being put to a logical high and being received by the mode latch via control line 363 (DATA-3), and then being put to a logical low which triggers the mode latch. The output of the mode latch 361 via control line 365 is then input to both the control switch buffer 357b (U7) via control line 359 (BUF) and the configuration memory 315 (U11) via control line 367 (RAM), and also to the memory decoder 345 by means of the further control lines 369a (RAM) and 369b (SW1/1) which are routed via a manually selectable switch being implemented as one of the switches of a DIP switch 371 which forms part of the manual controls 332. The mode latch 361 consequently performs a number of different control functions upon triggering by the control switch buffer 357b to effect switch-over, these control functions being summarised as follows:

- triggering a second latch being a page latch 373 (U10A) via control lines 367 and 377 (RAM), to be described in more detail later;
- switching over the configuration memory 315 via control line 367 (RAM) from a read mode to a write mode for writing to the zone controller 317 via internal bus line 375;

- 55 -

- triggering the memory decoder 345 via control line 369 (RAM, SW1/1) to lock out the configuration ports 347 via control lines 351 (CONFIG, PORT) to prevent the further writing of information to the configuration ports 347 via bus line 311.

The page latch 373 (U10A) in conjunction with associated decoder circuitry U17B, U17D and U17C shown in figure 11, effects operation of a paging system to conceal the code stored within the device ROM 343 (U9) after switch-over. Moreover, the flip flop U10A for the page latch 373 is configured in a toggle mode and is toggled by a combination of four states, the first being the issuance of a page swapping input signal received on the control line 378 (DATA-2) connected to one of the bus lines of the internal bus line 375b and sourced from the configuration memory 315 due to the previous disabling of the configuration ports 325; the second being the issuance of an output signal received on the control line 377 (RAM) from the mode latch 361 (U10B) which only allows the paging system to be invoked once the initialisation phase is complete; the third being that the current protection state is active, not being forced into a read write state by any corresponding manual operation of the manual controls 332 as signified by a control signal received on the control line 380 (CONTROL) from the zone controller 317; and the fourth being that the device ROM 343 is currently being addressed as signified by a control signal received on the control line 382 (ROM) from decoding circuitry U16A shown in figure 11 associated with the memory decoder 345. In this manner, once all four of the criteria have been met, the page swapping input signal drives the page latch 373, causing a toggle of the output signal issued thereby along the output 379 (PAGE) which is connected to an appropriate input of the device ROM 343 (U9), causing corresponding memory locations in different halves of the device ROM to

be alternately read from (i.e. paged), with successive read requests of the device ROM 343 issued by the computer via the bus interface 305.

In order to achieve this effect, the device ROM is divided into upper and lower halves, only one half of which (address lines A0 to A13 of the EPROM U9 as shown in figure 10, with A14 set high by PAGE) is accessed by the initialisation means during the power-up sequence of the computer system in the manner previously described, and the other half (address lines A0 to A13 of EPROM U9 with A14 set low by PAGE) containing unrelated information. This unrelated information can be used to (i) scramble the code stored within the one half after switch-over which occurs at the completion of the operation of the initialisation means if the EPROM U19 is directly read from at consecutive address locations by the CPU, and/or (ii) with appropriate decoding, can run executable code such as the terminate state ready (TSR) routine for the computer system.

The full definition for the driving of the page latch 373 is as follows:

- the head or track position of the disk drive 307 being disposed at a predetermined location as determined by the code contained within the device ROM 343 which is executed at the completion of the initialisation operation;
- the protection state as controlled by the manual controls 332 identifying the current disk location as being actively protected;
- the configuration ports 347 being locked out from further transfers of information from the bus 311 to

- 57 -

the configuration memory 315, by actuation of the mode latch 361; and

- a read request being initiated by the computer via the bus interface 305 to the defined operating window of the device ROM 343.

This has the effect of when all but the read request for the device ROM 343 matches, subsequent reading of the device ROM 343 causes the page latch 373 to select or page between the upper and lower halves of the two memory areas of the device ROM 343 every time an access is made to the device ROM. Thus, the code that is read from the device ROM comes from alternate halves of the ROM on consecutive accesses, which will scramble the code.

The third latch of the latch means 325 (U6B) is an audio latch 381 and the fourth latch is an update latch 383 (U6A). The flip-flops for these latches are also configured in the straight toggle mode and are respectively toggled by switches SW1 and SW2 of the buttons 385 as shown in figure 12, which also form part of the manual controls 332, via control lines 387 (AUDIO, UPDATE). The outputs of the respective audio and update latches 381 and 383 are connected via control lines 389 (AUDIOLED, UPD) to the zone controller 317 (U1) so as to form inputs which activate the manual override processor 327 to perform an overriding function of the protection scheme configuration stored within the configuration memory 315.

In the case of the audio latch 381, when a corresponding button 385 (SW1) is pushed, the manual override processor 327 overrides the normal disable configuration for the audio output of the display and audio output 391 by enabling it. This is achieved by the control line 389 (AUDIOLED) being connected to the reset input of a duration

- 58 -

extension timing circuit U5A associated with the zone controller externally of the FPGA U1, so as to enable the circuit, and by connecting the appropriate control line PROTECTED of the zone controller 317 to the trigger input of the timing circuit as shown in figure 17 of the drawings. Accordingly, whenever a violation of a protected area occurs a violation signal is output along the control line PROTECTED and a sound output signal issued at the output line SOUND of the timing circuit. This in turn is communicated via the connector J6 to the display and audio output 391 for sounding of an audio output to be described in more detail later.

In the case of the update latch 383, when a corresponding button 385 (SW2) is pressed, the manual override processor 327 overrides the normal zone configuration set for the user areas or zones of the disk drive, being defined for the particular storage location being accessed by the disk controller 303 at that particular time. This overriding forces the disk drive into a read/write state by allowing data writing to the accessed storage locations of the disk drive 307 which fall within the defined user areas or zones, regardless of the protection status previously defined for these particular user areas or zones. This is achieved by the output control line 389 (UPD) of the update latch 383 being connected to the processing circuitry of the read/write switch 333 and step switch 335 and utilising a control signal output from the processing circuitry of the switches via control line 360 (UPDAT) to disable the output of the even port 355 (U8). Consequently, the outputs of the zone definition information buffer 355a by virtue of their connection via pull up resistors (R8) to the supply line (VCC) are overridden, enabling the manual override processor 327 of the zone controller to operate accordingly. This facility is utilised, for example, when

updating software stored within a particular portion of the disk drive.

The manual controls 332 and the display and audio output 391 are actually implemented in a separate housing from the remainder of the protection system, with the exception of the DIP switches SW1 which are incorporated into the main card of the protection system. Communication between the manual controls and the display and audio output means 391 on the one hand and the protection system on the other is by way of a communication cable terminating in connectors J6 and J7 incorporated into the main card of the protection system and the housing of the manual controls and display and audio output respectively. As can be seen in figures 10 and 12, the same reference labels have been used to identify communication lines which are interconnected.

The manual controls 332 are designed to include other buttons 385 to allow for a physical override of the normal configuration stored within the configuration memory 315 to direct certain other functions to be performed in addition to the audio control and update control as previously described. Moreover a third reset button (SW3) is provided to reset the visual display of the display and audio output 391 after an attempted violation of the protection scheme configured for the disk drive has been detected by the protection system via one of the lines (CLEAR) of the control lines 388, which connect to the zone controller 317. Moreover, the display and audio output 391 has the ability to visually indicate when an attempt to violate the protection scheme configured for a particular zone is made. Furthermore, the reset button can allow for resetting of the visual display after indicating the attempt.

The remaining three buttons SW4, SW5 and SW6 function to provide for the manual selection of three levels of

- 60 -

protection. The button SW5, connected via another of the lines CLR100% of the control lines 388 to the zone controller 317, is for the defined user and system areas or zones as previously described which function as prescribed. The button SW6, connected via another of the lines MANUAL of the control lines 388 to the zone controller 317, is for providing read only protection for an entire hard disk. Finally the button SW4, connected via another of the lines AUTO of the control lines 388 to the zone controller 317, is for an intermediate level of protection between the previous two which allows the user to work with the level of protection provided by SW5, but upon detecting an attempted violation of the protection scheme, automatically switches the level of protection to the level provided by SW6, whereby the entire hard disk is protected.

As shown in figure 15 of the drawings, all of the push buttons SW1 to SW6 are connected respectively to a bank of pull up resistors and capacitors for debouncing and biasing purposes.

The display and audio output 391 as shown in figure 12 comprises a visual display feedback consisting of a series of LEDs L1-L9 and an audio output consisting of a piezo electric beeper B1. The visual display relates current conditions within the selected disk drive and include a drive select LED L7 connected via line DSEL, a protection level LED L9 connected via line LED 100, an audio state LED L8 connected via line AUDIOLED, and a set of condition LEDs to represent all possible protection zones, namely the system zone L6 (lines RED0, GREEN0), user zone A L5 (lines RED4, GREEN4), user zone B L4 (lines RED5, GREEN5), user zone C L3 (lines RED6, GREEN6), user zone D L2 (lines RED7, GREEN7) and one other zone L1 (lines RED100, GREEN100) representing undefined locations on the drive.

- 61 -

The drive select LED L7 simply indicates when the selected drive becomes active by being switched on or off. The control line DSEL therefor is sourced via inverting logic U4B as shown at figure 11 from the CLK_ENABLE line which is connected to appropriate processing circuitry provided within the FPGA U1 and the PAL U2 which respectively decode control signals provided on the disk controller disk drive interface 309 and the computer bus interface lines 311 which indicate drive selection.

The protection level LED L9 simply indicates which of the aforementioned protection levels set by the aforementioned buttons SW4 to SW6 apply to the relevant disk drive. The control line LED100 therefor is sourced directly from processing circuitry for the zone controller provided within the FPGA U1. The FPGA uses appropriate processing circuitry to drive the LED L9 in a manner to indicate which of the three levels of protection are provided. Moreover, the three levels are represented by the LED L9 being switched on, off or alternatively being flashed. In order to generate the flashing signal, the FPGA U1 uses an external oscillator circuit U5B as shown in figure 16 to generate an oscillating signal which is input to the FPGA U1 by way of control line FLASH. Accordingly the FPGA U1 switches the oscillating signal input FLASH to be output at the LED100 line to flash the LED L9 when appropriate.

The audio state LED L8 simply indicates whether the audio outlet has been enabled or disabled by being switched on or off. The control line AUDIOLED therefor is sourced from the audio latch 381 to activate the LED L8 at the same time as activating the duration extension timing circuit U5A as shown in figure 17 of the drawings for generating the audio driving signal input to the piezo electric beeper B1 along control line SOUND as shown in figure 12.

- 62 -

The condition LEDs are tri-colour LED's each have four states as follows:

- (a) Off - Not the currently accessed part of the drive which is not defined for protection.
- (b) Green - Not the currently accessed part of the drive, but is defined to be protected.
- (c) Amber - Currently doing a valid access function.
- (d) Red - Violation detection in this zone.

The processing for generating the control signals for driving the condition LEDs L1 to L6 and the beeper B1 is performed by the logic for the zone controller 317 and the state set by relevant buttons 385 of the manual controls 332. Importantly, in order to determine an attempt at violating a protected zone of the disk drive, the zone controller 317 needs to receive relevant control signals which are fed back from the read/write switch 333 and also the step switch 335 as a result of either of these switches having to perform an overriding control function as dictated by the zone controller 317. Consequently appropriate control lines such as PROTECTED, CONTROL, CLK_ENABLE, CLK_P, PL, DIRECTION, DATA_1 etc are interconnected between the FPGA U1 and the PAL U2 for communication of relevant signals to enable these and other analyses to be performed.

As previously described, the manual controls 332 include a DIP switch 371. The DIP switch 371, implemented as SW1 in figure 10, comprises a set of switches which perform three basic functions:

- (a) enabling or disabling of the controlling functions of the protection device;
- (b) defining the location within the addressable range of the CPU to place the device ROM window;
- (c) defining which physical hard disk drive to protect by selecting the appropriate drive label.

With respect to enabling or disabling the controlling functions of the protection device, when the corresponding switch (switch 1) is pressed to the off position, a disabling control signal is issued by a control line 395 (SW1/1) to the zone controller 317 (U1) by virtue of the connection of the control line 395 (SW1/1), via a pull up resistor of the bank of pull up resistors (R8), to the supply line (VCC) of the protection system, disabling the zone controller from issuing overriding control signals for the read/write switch 333 and the step switch 335 (U2). When switch 1 is pressed to the on position, an enabling signal is provided along the control line 395 (RAM, SW1/1) to enable normal operation of the zone controller 317 (U1).

This enabling or disabling of the zone controller 317, and hence the controlling functions of the protection device, is also effected automatically prior to the device ROM being run to allow for the configuration memory to be loaded prior to it being used. Moreover, switch 1 of the DIP switch is connected to the output control line 369a (RAM) sourced from the output of the mode latch 361 via control line 365, so that when the switch is in the on position, a disabling signal for the zone controller 317 can nonetheless be issued by the mode latch 361, disabling the zone controller from issuing overriding control signals, at least during the initialisation phase. Thus during each cold boot or power up of the computer system,

- 64 -

the output of the mode latch 361 is reset via control line 399 (PWRRESET), providing a reliable state disabling the overriding control of the zone controller 317 until configuring of the configuration memory 315 during initialisation has been completed. Thereafter, the function which effects switch over, also initiates protection when the output state of the mode latch 361 changes, generating an enabling signal along line 395 (SW1/1) for the zone controller.

The defining of the location of the device ROM window within the addressable range of the CPU is effected in the usual manner by simply setting the appropriate binary code within prescribed switches for the valid address locations of the device ROM window, which are simply output along address lines 397 connected to the memory decoder 345. In this embodiment, three switches (switches 2, 3, 4) of the DIP switch 371 are utilised which are connected at one side via the network of pull up resistors (R8) to the supply line (VCC) of the protection system, and at the other side to the memory decoder 345 via the enable line 397 (GND). The signals provided by these three switches produce three states, one state per switch, each used as part of the address definition which is decoded by the memory decoder. As shown in figure 10 of the drawings, in the present embodiment the three switches are all connected to a common network so that all three of the switches need to be on or closed in order to enable the device ROM.

Defining the particular hard disk drive for protection is performed simply by setting the corresponding switch for the drive to the on state. In this embodiment, provision is made for two disk drives by two switches (switches 5, 6). One side of each switch is connected to the respective main input lines 309 output by the controller 303 which select the appropriate disk drive, via input lines 400 (SELECT_1,

SELECT_2), and the other side of each switch is connected to the zone controller 317 via a corresponding input line 401 (CLK_ENABLE) and to the address generator 313 via a corresponding input line 403 (CLK_ENABLE). Thus, the zone controller 317 is made cognisant of the particular drive being accessed to direct the necessary overriding processing and zone section analysis for the particular drive, along with the address generator so that the appropriate memory location for the particular disk drive is accessed by the pointer. Hence, if physical drive 1 was to be protected and the others not, switch 5 would simply be set on and the switches for the remaining drive(s) would be left in the off or open state. Hence, as the controller 303 accesses this disk, the appropriate control line (SELECT_1) will be clocked, which in turn will clock the relevant processing circuitry of the FPGA U1, and also the relevant circuitry of the PAL U2. The last of the remaining switches switch 8 has one side directly connected to the particular control line PROTECTED output by the zone controller to signify when a particular violation has occurred, and the other side of the switch is connected directly to the WRITE_FAULT control line of the disk controller 303, so that when the switch is closed or on, the controller is alerted to the occurrence of a write fault to the disk drive 307, as a result of an attempted write to a protected area.

The remaining switch, switch 7 is unused in the present embodiment.

Now describing the method of operation of the initialisation means with reference to the operation of an IBM PC or compatible, as previously described, the computer system as part of its start up sequence, performs a test for all device ROMs in memory. Accordingly, the protection system obtains control of the computer system during the

start up or power up sequence by having a device ROM in an appropriate addressable location that conforms to the standards for a definition of a device ROM. However instead of simply providing a signature code as part of the BIOS extension ROM, in the present embodiment the device ROM actually goes through a loading procedure or configuration set up for implementing the protection scheme to be implemented for the particular disk drive connected to the computer system. It should be noted that this was not the original purpose intended for such ROMs, as they were to be a way of supplying extensions to the BIOS table, in order to update the table of routines for communication with all fitted hardware in the particular computer system. However, there is no problem with operating the device ROM 343 of the present protection system 301 in this manner, as long as the normal operation of the power-up sequence is not affected in any way.

The power-up sequence results in a predefined set of conditions taking place in the computer and accordingly the initialisation means works in conjunction with this by setting parts of the protection system into a state suitable for receiving information and allowing the protection scheme to be configured for the disk drive. This is effectively done in two parts, the first using the master reset signal (BRESET) from the bus interface 305 which is connected directly to the latching means 325 via an inverter U4D and the control line 399 (PWRRESET) to load a predefined state into the latching means; and the second is to place the device ROM 343 into one of the 2 Kbyte boundaries of the memory map of the computer system.

As previously described, the setting of certain switches of the DIP switch 371 and the memory decoder 345 place the code of the device ROM into a valid address of the CPU of the computer. Thus, upon the CPU addressing the appropriate

- 67 -

2 Kbyte boundary during power-up and ascertaining the presence of the device ROM 343, relevant code of the device ROM is executed by the CPU enabling the device ROM to take over operation of the computer system at this power up stage so that the initialisation means of the protection system can operate. The sequence of operation of the initialisation means, as previously described, commences with running the CMOS validity test and correction program to ensure that the main memory and peripherals of the computer system are operational and can be used by the protection system.

Next the password filter program is operated so that if appropriate information is being stored for user passwords, the program can operate the computer to interact with responses from the user of the computer system via the screen and keyboard thereof. This allows for entry of user information and passwords of any forms and execution of this operation can be logged to a predefined part of the hard disk to be used as a data log which would indicate what happened when and by whom.

Depending upon the particular user which has been logged into the computer system, the virtual environment switching program is operated to define a virtual operating environment for the particular user, defining protection levels for zones which can or can not be accessed by the user. These zone definitions are conveyed by the device ROM 343 to the even port of the configuration ports 347 to subsequently be combined by the zone section processor 329 of the zone controller 317 for overriding the appropriate drive control signals by the read write and step switches 333 and 335 respectively at the appropriate times.

The next stage in the sequence of operation of the initialisation means is to load the appropriate protection

- 68 -

definition for the protection scheme into the configuration memory 315. This is done by a sequence involving the CPU addressing the hard disk controller 303 to move the head of the hard disk to a physical position corresponding to a particular storage location of the disk drive, which action in itself is interpreted by the address generator 313 to point to a corresponding memory location within the configuration memory 315 which corresponds to the storage location currently accessed by the head of the disk drive. Simultaneously with this, the CPU writes appropriate status information to the odd port 357 of the configuration ports 347 and in particular to the configuration definition buffer 357a, which status information represents how to treat this storage location. This information is subsequently written to the address of the configuration memory 315 pointed to by the address generator 313 by operation of the memory decoder 345. This moving and writing sequence is then cycled through for consecutive storage locations, until all valid storage locations of the hard disk have been defined. In the park track, encoded status information, which functions as a trigger for the switch over operation of the initialisation means is stored within the corresponding memory location of the configuration memory.

The configuration load sequence is achieved by using the bus interface 305 to communicate data from the CPU of the computer through to the odd port 357 which holds the latch data until it is reloaded. The output of the odd port is in the normal mode state and not in the high impedance mode provided by the tri-state buffers. The odd port is set up in this way as a result of the mode latch 361 being reset from the bus interface 305 when the cold boot was initiated, as previously described. The mode latch 361 also maintains the configuration memory data transfer port 330 in the read mode so that the status information written to

the configuration definition buffer 357a is latched directly into the memory location of the address pointed to by the address generator 313.

The address generator 313 generates the pointer address as a result of receiving two groups of driving signals from the drive interface 309. The first is the head select signals which come almost directly from the controller cable of the drive interface 309 and the second is from the signals that direct the hard disk mechanism to move the arm that the heads of the disk drive are mounted on. This second group drives into the counter 323 which can count either up or down. Consequently, the address generator keeps an accurate account of the track location selected by the heads.

By moving the drive arm of the disk drive in any direction, the address pointed to by the address generator for the purposes of the configuration memory is correspondingly changed. In this manner, the co-ordinate to the configuration memory is easily selected by moving the disk drive to the location to be programmed and putting the required value into the odd port. This has the effect that if the head is at track zero head zero and the odd port has the status information 00 in it for example, the memory location of the configuration memory that relates to this particular co-ordinate will store 00 in it.

There is however, a complication arising from this, in that if the only thing that changes is the movement of the head to the next track, the status information 00 from the odd port will be stored into the location for the next track ie; track 1/head 0. Although this is convenient in that it is not necessary to reload the port for each location, if the hard disk controller has automatic retries built into it, which is often the case, and it decides to reset its

- 70 -

internal counters in the event that it has trouble locating a sector (a situation occurring not infrequently), then the entire contents of the memory from track 0 to whichever track it was seeking is filled with the binary value for the status information contained in the odd port. In order to accommodate for this, the address sequence for the configuration memory commences with the track furthest from track 0 and works backwards towards track 0. Thus, if a seek to track 0 occurs, it will not overwrite any of the previously configured memory.

Once the configuration memory 315 has been configured, the initialisation means commences a switch over operation to lock out any further writing to the configuration memory 315 via the configuration ports 347, so as to prevent any tampering by unknown sources. The switch over is initiated by the mode latch 361 being written to via the control switch buffer 357b of the odd port, whereby a data bit is used as a flag which must be put to a logical high and then to a logical low. The transition from high to low triggers the switch over and commences the run time mode of the protection device.

In the run time mode, the protection device has been designed to provide a minimum overhead on the normal functions of the computer system. Thus, the key functions performed by the protection device in the run time mode are the monitoring function performed by the address generator 313, configuration memory 315 and zone controller 317 and the controlling or switching function which is performed by the zone controller 317 and the read write switch 333 and step switch 335.

Dealing firstly with the monitoring function, the address generator 313 is used to monitor the hard disk controller 303 and process output signals therefrom into an address to

be pointed to within the configuration memory 315 which corresponds to the storage location intended to be accessed by the disk controller 303. As opposed to the operation of the protection system during the initialisation means, during the run time mode, the configuration memory 315 is switched to operate in the read mode by virtue of the mode latch 361, whereby status information data stored at the memory location of the address pointed to by the address generator 313 is output from the data transfer port 330 and received by the zone controller 317, this data transfer port 330 previously being used for inputting status information into the configuration memory during the initialisation or set up phase. The status information data which is output by the configuration memory is not only read by the manual override processor 327, but is also monitored by the mode and page latches 361 and 373 respectively.

During the run time mode, the mode latch 361 is locked into a state that will not alter until a reset signal is detected along the reset control line 399 from the bus interface 305. Consequently the issuance of a reset signal would not occur again without the computer system going through a cold boot and hence re-running the initialisation means for setting up the protection system during the power up sequence of the computer. Accordingly, the input to the mode latch from the internal data bus 375b is only utilised in practice during the operation of the initialisation means. The page latch 373 receives an input from the internal data bus 375b to receive a bit of stored status information output by the configuration memory 315 for the memory location being pointed to, to indicate whether the storage location being accessed by the controller 303 at that particular point in time is one for which page swapping, as previously described, is to be performed. Consequently the page swapping signal input to the page

- 72 -

latch 373 via control line 377 effects toggling of the page latch 373 to encrypt the code stored within the device ROM 343 in the manner previously described.

With the configuration memory 315 producing the conditions to be performed at the current drive location, the remainder of the status information data is used to drive the zone controller processors in the manner previously described. After performing appropriate manual override controls and combining zone section information by the manual override processor 327 and zone section processor 329 respectively, the zone analysis processor 331 produces the eventual controlling set of signals which in turn drive the read write switch 333 and step switch 335 in the manner previously described. In addition, the zone analysis processor 331 drives the display and audio output 391 in a manner dependent upon the result of its analysis and the state of the manual controls 332.

It should be appreciated that the utility of the protection system can be extended well beyond protecting the computer system from computer viruses, and that quite significant enhancements can be made to the overall design and functionality of computer systems by the ability of the protection system to secure any portion of the disk drive from reading or writing in accordance with the desired configuration of the system. Indeed, the inclusion of the bus interface 305 and the ability of the computer to read the code stored within the device ROM 343 during the power-up boot sequence can be used to great advantage. As previously indicated, code executed during accessing of the device ROM 343 during power-up need not be limited by being physically stored upon the device ROM 343, whereby the device ROM 343 can in fact be designed to read further code from the hard disk and drive.

- 73 -

The code within the device ROM can also be programmed so that the device ROM instructs the CPU to read further code from the hard disk drive 307 for the purpose of using the code stored upon the hard disk drive to set up the configuration of the configuration memory 315. This programming scheme allows for the smallest amount of space provided on the device ROM to be used whilst still allowing for as much code to be run as is necessary which code is physically stored on the hard disk drive to operate the initialisation means.

It should be noted that the storage locations being read from the hard disk by this device ROM code during the boot sequence is intended to be relevant to the boot sequence only. Hence any requests by the computer to read from these same storage locations after the boot sequence is completed would be unjustified. This being the case, the protection system can configure these storage locations on the hard disk so as not to be read from during the normal operation of the computer system.

As the configuration of the configuration memory 315 is performed by the initialisation means, pursuant to the running of this code, the code can be stored in a protected area of the disk drive. That is, the running of the code precedes the normal operation of the protection device, which will perform the requisite protection of the designated storage areas of the disk drive during normal operation of the computer after the boot sequence has been completed. Thus there is no difficulty in read protecting this particular storage location which stores the relevant code for the device ROM.

As previously described, the code stored within the device ROM or within the hard disk for the device ROM can also be programmed to protect user and/or password data within a

- 74 -

secure location on the hard disk, instead of within the main memory of the computer system. In this respect, access could be restricted at both read and write levels, while still allowing the boot sequence to have normal access for password validation and log data.

Furthermore, the ability to control both reading and writing of data from or to the hard disk allows for the configuration of the system to be personalised to the particular user logging on. For example, if user A logs on, he would automatically be given an environment that was predefined a set way. If user B logs on a different set of configurations is loaded and the environment is set to their predefined configuration. This could use not only read and write controls but the partition data and allow for adjustment of the hardware and software configuration of the system as predetermined by the code run by the initialisation means.

As can be seen from the above, the versatility of the design of the protection system and the increased functionality and flexibility that can be provided to the computer system by virtue of adopting the protection system can be quite profound, opening up the possibility of improving the design of computer systems quite markedly.

One of the differences between the device, system and method of this invention as described in the preceding embodiments and other known virus protection products, is that the presence of the protection device ensures that the computer operating system is clean and the computer is able to power-up when the computer is switched on or undergoes a cold boot. Known virus protection products, be they hardware based or software based rely on the computer having a clean operating system before they can perform their task. There are specific advantages in the preceding

embodiments insofar as the protection device is concerned, being intended for interposition between the hard disk drive and the disk controller, in that it does not take up a bus-slot in the computer (first embodiment only), it is easy to install, and by virtue of its operating method, does not reduce the operating speed of the computer.

The protection device has a further advantage insofar as it prevents a hard disk drive from inadvertently being reformatted. In addition, it prevents viruses from trying to perform a low level format of a hard disk. The protection device, by causing a read only function to the protected area of the read write store would cause any low level format to abort due to the protected portion of the hard disk drive not being reformatted, and resulting in an initialisation failure.

It should be appreciated that the scope of the invention should not be limited to the scope of the specific embodiment described herein. In particular, utilising the form shown in figure 1, a protection device for protecting sensitive areas of a floppy disk for use in an IBM PC or compatible, separate from or additional to a hard disk is also envisaged within the scope of the invention.

THE CLAIMS defining the invention are as follows:-

1. A protection system for preventing corruption of computer software instructions relating to the power-up sequence instructions of a computer associated with a read write store comprising:-

(i) a protection device for protecting a predetermined portion of the read write store, said protection device connected between the read write store and the computer and comprising: input means to receive address information from the computer; means for determining when the computer is addressing the predetermined portion from the address information; and means for disabling the read write store from being written to, or alternatively from being read from, depending upon the type of protection being provided, when said means for determining determines that said computer is addressing the predetermined portion; and

(ii) a storage location provided within the predetermined portion of the read write store for storing the computer software instructions;

wherein said protection device is adapted to prevent the computer software instructions from being altered by preventing the predetermined portion from being written to.

2. A protection system as claimed in claim 1, wherein the computer software instructions include system configuration information.

3. A protection system as claimed in claim 2, wherein the system configuration information includes a partition table.

- 77 -

4. A protection system as claimed in any one of the preceding claims, wherein the predetermined portion includes further storage locations for storing system operation information.

5. A protection system as claimed in claim 4, wherein the system operation information includes disk operation system information.

6. A protection system as claimed in claim 5, wherein the disk operation system information includes boot sector information.

7. A protection system as claimed in any one of the preceding claims, wherein the system includes a copy of system operation information stored within the predetermined portion, copied from an unprotected portion of said read write store when said system is installed in the computer, and means whereby upon power-up or cold boot of the computer a copy of the copy of system operation information within the predetermined portion is copied to the unprotected portion for subsequent use in the power-up sequence, thereby ensuring the system operation information is uncorrupted at least at power-up or at a cold boot of the computer.

8. A protection system as claimed in any one of the preceding claims, wherein the predetermined portion is adapted to contain a cyclic redundancy check number for each of one or more computer files contained in other unprotected portions of the read write store, whereupon power-up or cold boot of the computer said means is adapted to calculate a fresh cyclic redundancy check number for each of said one or more computer files and compare said fresh cyclic redundancy check numbers with respective the cyclic redundancy check numbers to provide an indication in

- 78 -

the event of any one of the one or more computer files being corrupted.

9. A protection system as claimed in claim 8, wherein the indication includes information identifying the computer file which has been corrupted.

10. A protection system as claimed in claim 8 or 9, wherein the one or more computer files includes hidden system files and a command interpreter file, the computer being an IBM PC or compatible thereof running DOS software.

11. A protection system as claimed in any one of the preceding claims, wherein said means for determining when the computer is addressing the predetermined portion includes decoder means for providing a flag when the address information corresponds to the predetermined location and said means for disabling includes control means being activated to prevent the read write store from being written to, or alternatively from being read from, as appropriate, in response to said flag when the predetermined portion is being addressed.

12. A protection system as claimed in claim 11, wherein said control means is interposed between the computer and the read write store to intercept selected address information output by the computer and selectively override the selected address information to disable the read write store from being written to, or alternatively from being read from, as appropriate.

13. A protection system as claimed in any one of the preceding claims, wherein said input means is connected to the control lines between the computer and the read write store to monitor address information being output by the computer to the read write store.

14. A protection system as claimed in any one of the preceding claims, wherein the read write store is a magnetic disk and the predetermined portion comprises a continuous section of storage area on the magnetic disk of a minimum size of at least one track of one head.

15. A method for preventing computer software instructions relating to the power-up sequence of a computer system comprising a computer and a read write store, from being corrupted by software corrupting instructions, the method comprising:-

providing a read only storage zone in a predetermined portion of a read write store for containing the computer software instructions;

protecting said predetermined portion of the read write store by: (i) monitoring address information of the read write store from the computer; (ii) determining when the computer is addressing said predetermined portion from said address information; (iii) and disabling the read write store from being written to, or alternatively from being read from, depending upon the type of protection being provided, when determining that the computer is addressing said predetermined portion; and

copying at least some of the computer software instructions into another portion of the read write store upon power-up or cold boot of the computer.

16. A method as claimed in claim 15, wherein the determining of when said computer is addressing said predetermined portion includes decoding the address information and providing a flag when said address information corresponds to said predetermined location.

17. A method as claimed in claim 15 or 16, wherein the disabling of the read write store is effected by selectively overriding control of the operation of said read write store in response to said flag by causing said predetermined portion only to be read when said predetermined portion is being addressed.

18. A method as claimed in any one of claims 15 to 17, including the step of verifying the status of one or more computer files contained in the unprotected part of said read write store, comprising calculating a fresh cyclic redundancy check number for each of said one or more computer files and comparing said fresh cyclic redundancy check number against a cyclic redundancy check number previously determined for each of said one or more computer files and stored in said predetermined portion, in order to determine if any one of said one or more computer files has been corrupted by corrupting-software instructions.

19. A method as claimed in claim 18, including the step of displaying details of any computer files where a variation is found between said cyclic redundancy check number and said fresh cyclic redundancy number, and/or instigating appropriate action directed by the user.

20. A method for preventing corruption of computer software relating to and associated with the power-up sequence of a computer communicating with a read write store comprising:-

storing code in a storage location of said read write store, which location is accessed by said computer during the initial power-up sequence of said computer to invoke a prescribed subroutine initially upon every power-up or cold boot of said computer;

- 81 -

allocating a predetermined portion of said read write store, including said storage location for protection;

storing a copy of fixed computer software instructions and information essential to the operation of said computer and which is also accessed by said computer during the initial power-up sequence, but which is located outside of said predetermined portion, within said predetermined portion;

storing said subroutine within said protection portion for invoking upon every power-up or cold boot of said computer;

performing a reference check on the validity of prescribed system files containing essential instructions and information for the operation of the computer and read write store; and

storing relevant reference information related to said check within said prescribed portion;

wherein said subroutine causes: (i) a copy of said fixed computer software instructions and information to be written over the existing said fixed computer software instructions and information located outside of said predetermined portion; (ii) a rechecking of the validity of said prescribed system files by performing a fresh reference check and comparing the relevant information obtained therefrom with the original said reference information stored within said prescribed portion; and (iii) performing contingency action if said prescribed system files are found to be invalid.

21. A method as claimed in claim 20, including the step of checking the presence of a protection device within a protection system of the type claimed in any one of claims

- 82 -

1 to 18 connected to said computer and read write store before relinquishing control of said computer.

22. A method as claimed in claim 20 or 21, wherein said storage location contains the partition table for said computer system.

23. A method as claimed in any one of claims 20 to 22, wherein said fixed computer software instructions and information comprises boot sector information.

24. A method as claimed in any one of claims 20 to 23, wherein said prescribed system files comprise hidden disk operating system files and a command interpreter file.

25. A method as claimed in any one of claims 20 to 24, wherein said reference check is a cyclic redundancy check and said relevant reference information is a cyclic redundancy check number.

26. A protection system for protecting a computer system comprising a computer and a read write store, the system comprising:-

a protection device for protecting a predetermined portion of the read write store, said protection device comprising: input means to receive address information from said computer; means for determining when the computer is addressing said predetermined portion from said address information; and means for disabling the read write store from being written to, or alternatively from being read from, depending upon the type of protection being provided, when said means for determining determines that the computer is addressing the predetermined portion; and

an initialisation means for operating the computer system during the power up sequence performed from a cold boot of the computer system to set up the predetermined portion of the read/write store for subsequent protection by said protection device during the normal operation of the computer system.

27. A protection system as claimed in claim 26, wherein said initialisation means operates the computer system to perform the method as claimed in any one of claims 20 to 25.

28. A protection system as claimed in claim 26 or 27, wherein said initialisation means is adapted to initially configure said protection device in accordance with a prescribed protection scheme for the read write store.

29. A protection system as claimed in any one of claims 26 to 28, wherein said means for determining when said computer is addressing said predetermined portion includes decoder means for providing a flag when said address information corresponds to said predetermined location and said means for disabling includes control means being activated to prevent said read write store from being written to, or alternatively from being read from, as appropriate, in response to said flag when said predetermined portion is being addressed.

30. A protection system as claimed in claim 29, wherein said control means is interposed between the computer and said read write store to intercept selected address information output by the computer and selectively override said selected address information to disable said read write store from being written to, or alternatively from read from, as appropriate.

31. A protection system as claimed in any one of claims 26 to 30, wherein said input means is connected to the control lines between the computer and said read write store to monitor address information being output by the computer to said read write store.

32. A protection system as claimed in claim 31, wherein said input means is also connected to the main communication bus of the central processing unit (CPU) of the computer to allow for implementing a protection scheme which selectively protects different predetermined portions of said read write store.

33. A protection system as claimed in any one of claims 26 to 32, wherein said means for determining includes a configuration memory comprising a look up table having a series of memory locations corresponding to all of the defined storage locations of said read write store, said memory locations including status information defining said protection scheme for said read write store, an address generator connected to said input means for generating a pointer to said configuration memory, said pointer corresponding to the address of the memory location of said configuration memory which corresponds to the particular storage location of said read write store being accessed by the computer at any prescribed time, and controller means for receiving the status information of the memory location being pointed to by said pointer and analysing said status information to determine whether the computer is addressing a said predetermined portion of said read write store.

34. A protection system as claimed in any one of claims 26 to 33, wherein said means for disabling includes a read/write switch and a step switch, whereby said read/write switch is adapted to prevent said read write store from being written to, or alternatively read from as

appropriate, and said step switch is adapted to maintain synchronicity between the storage location actually being accessed by the computer and the storage location being determined by said protection device, in accordance with the analysis of said status information performed by said controller means.

35. A protection system as claimed in any one of claims 26 to 34, wherein said initialisation means includes a further memory connected to that part of said input means which connected to the main communication bus of the CPU, said further memory containing executable code to operate said initialisation means, a decoder for controlling communication with said further memory, a communication port connected between said further memory and said configuration memory to buffer communications therebetween, and latching means associated with said decoder, said communication port and said configuration memory to control the sequence of operation of said initialisation means and protection device such that said initialisation means is operated initially to configure said configuration memory in accordance with said protection scheme, and then the protection device is operated to provide ongoing protection during normal operation of the computer system in accordance with said protection scheme.

36. A protection system as claimed in claim 35 as dependent upon claim 33, wherein said initialisation means causes (i) said address generator to generate said pointer for an address of said configuration memory at one end corresponding to one end of the storage locations of said read/write store, (ii) the status information for the pointed address of said configuration memory to be loaded into said communication port for writing to said configuration memory, and (iii) said address generator to consecutively point to successive addresses locations

whilst continuously writing to said loaded status information into said configuration memory, said status information being changed to suit the particular protection scheme for the corresponding storage locations of the read/write store when necessary.

37. A protection system as claimed in claim 36, wherein said one end is the last storage location of said read/write store and said address generator points to successive address locations working backwards from said one end.

38. A protection system as claimed in any one of the preceding claims as dependent upon claim 35, wherein said latching means enacts a switch over between the operation of said initialisation means and said protection device to lock out further communications with said configuration memory and said communication port via the main communication bus of the CPU upon completion of the operation of said initialisation means.

39. A protection system as claimed in any one of claims 35 to 38, wherein said latching means pages between separate halves of said further memory in response to consecutive read accesses of said further memory by the computer after said switch over to encode information stored in either half of said further memory.

40. A protection system as claimed in any one of claims 26 to 39, wherein the protection system includes manual controls to manually override certain status information of certain memory locations of said configuration memory if desired.

41. A protection system as claimed in any one of claims 26 to 40, wherein the protection system includes a display and

- 87 -

audio output for providing a visual and/or audio output indicating the status of said protection device during operation thereof.

42. A method for protecting a computer system comprising a computer and a read write store, the method comprising:-

setting up a predetermined portion of the read write store during the power up sequence performed by the computer from a cold boot for protection during the normal operation of the computer system; and

protecting said predetermined portion by: (i) monitoring address information of the read write store from the computer; (ii) determining when the computer is addressing said predetermined portion from said address information; (iii) and disabling the read write store from being written to, or alternatively from being read from, depending upon the type of protection being provided, when determining that the computer is addressing said predetermined portion.

43. A method for protecting a computer system comprising a computer and a read write store, said read write store being configured so that information stored thereon is divided into a plurality of different zones having a prescribed status, each of the zones being further divided into a plurality of different sectors in which the information is stored, said method comprising continuously controlling the accessing of storage locations of said read write store by the computer during the operation thereof, without interfering with the normal operation of the computer, depending upon the type of information stored within a sector and the type and status of the zone within which the sector is located as determined by the configuration of the read write store; wherein said controlling includes allowing, restricting or preventing

- 88 -

reading or writing operations by the computer from or to the read write store.

44. A method as claimed in claim 42 or 43, including: continuously accessing discrete memory locations of a configuration memory defining the configuration of storage locations of the read write store, simultaneously with the accessing of these storage locations by the computer of the computer system; and analysing the configuration definition for the accessed storage location as accessed within the configuration memory to determine the controlling to be performed.

45. A method for protecting a computer system as claimed in claim 44 including: obtaining control of the computer system during its power up sequence performed pursuant to a cold boot; and configuring the configuration memory for subsequent access.

46. A method as claimed in any one of claims 43 to 45, wherein the configuration memory has a series of memory locations corresponding to the defined storage locations of the read write store of the computer system, within which the definition of a protection scheme for said storage locations is stored.

47. A method as claimed in claims 45 or 46, including locking the configuration memory from further configuration by the computer after completing the initial configuration thereof, such that only read accesses to the configuration memory are allowed during the normal operation of the computer system.

48. A protection system substantially as herein described in any one of the embodiments.

49. A method for preventing computer software instructions relating to the power-up sequence of a computer from being corrupted as herein described in any one of the embodiments.

50. A method for protecting a computer system substantially as herein described in any one of the embodiments.

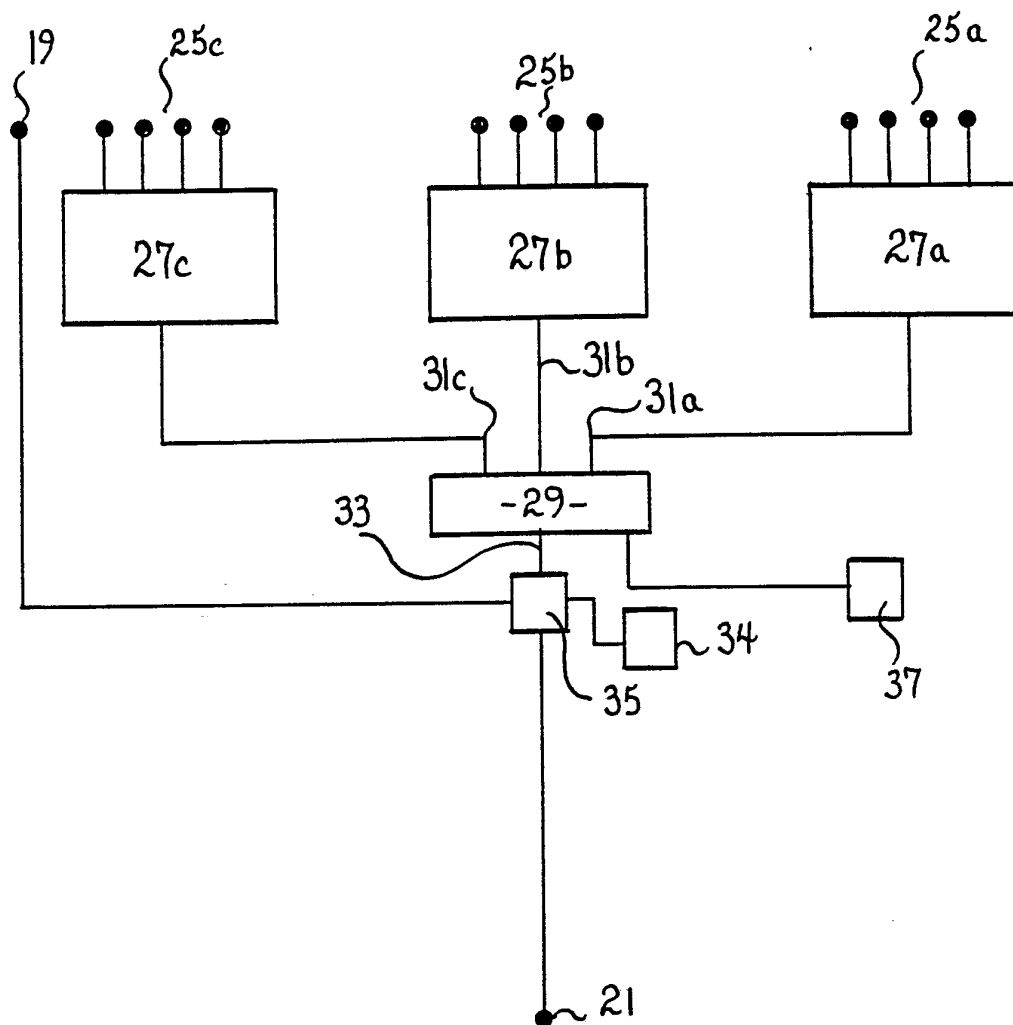


Fig. 1

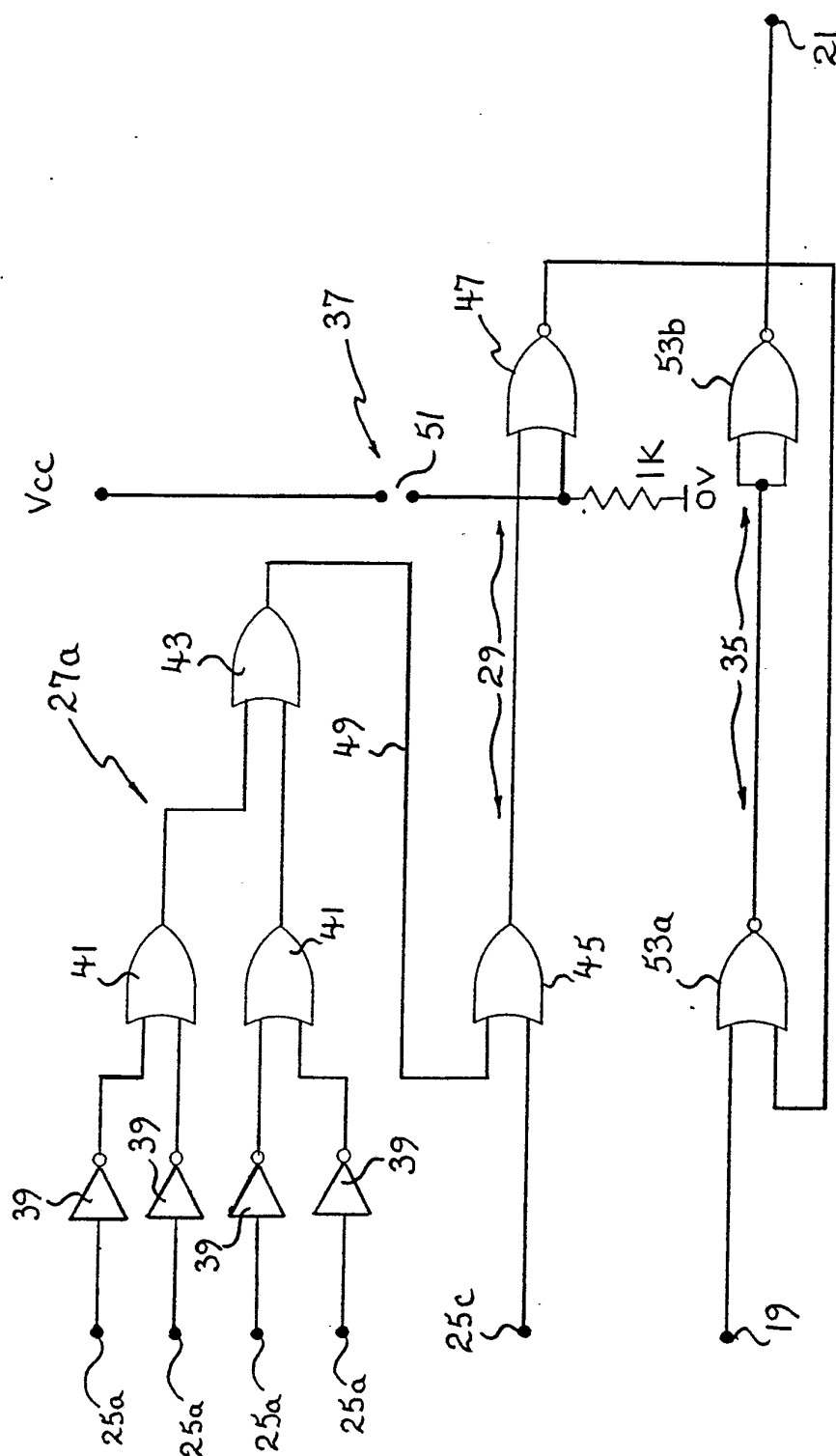
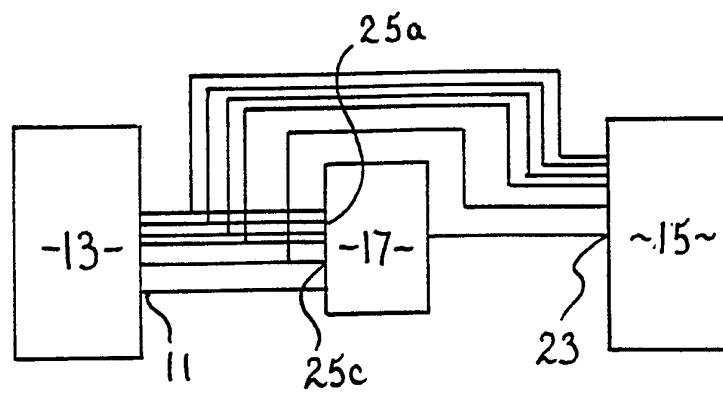
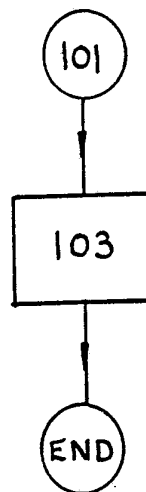


Fig. 2

3/18

**Fig. 3.****Fig. 4.**

4/18

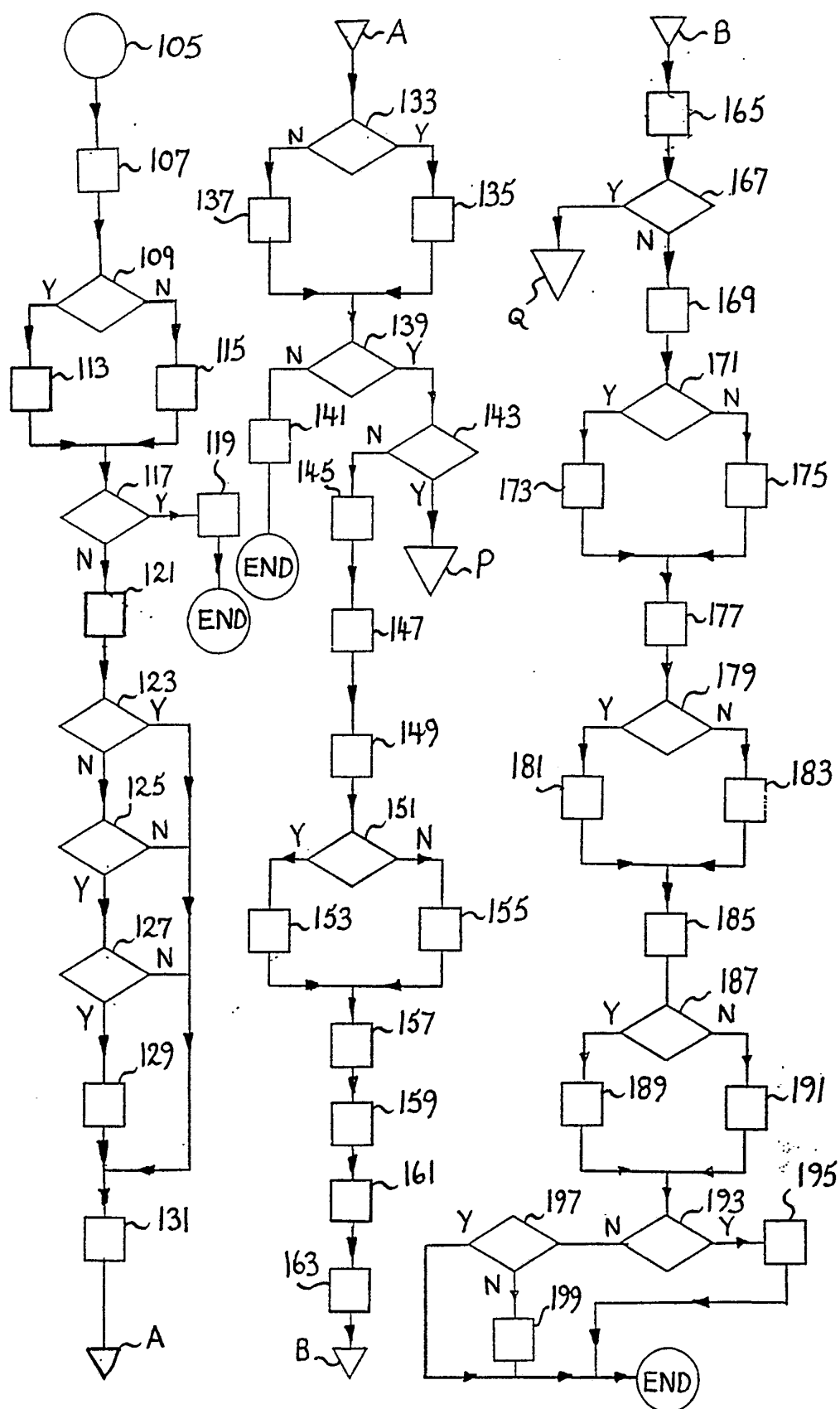
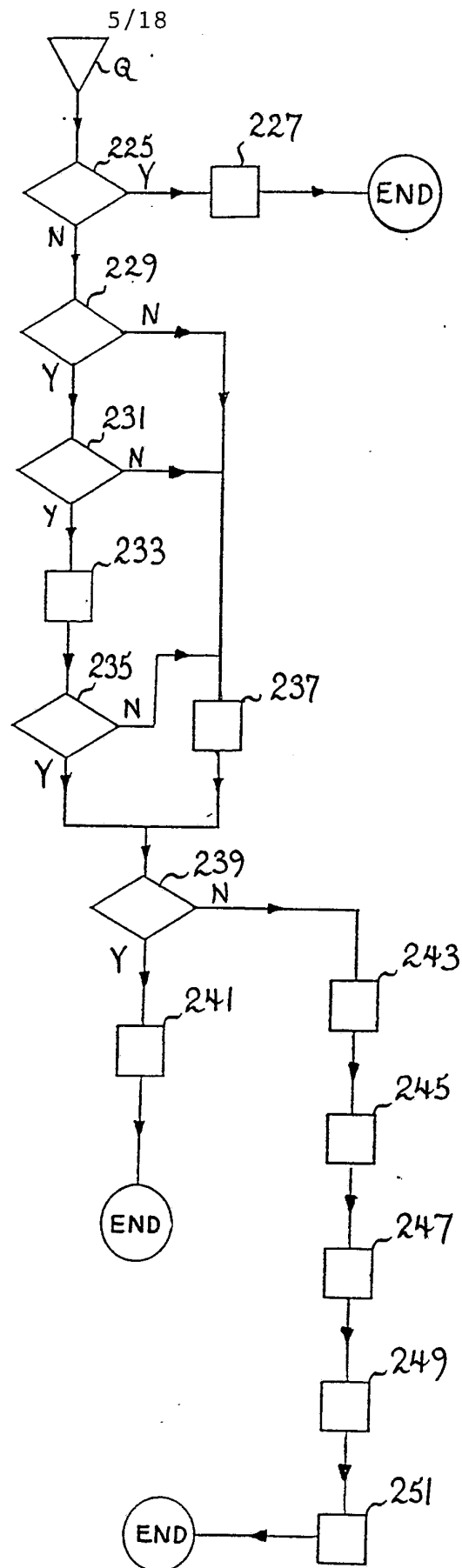
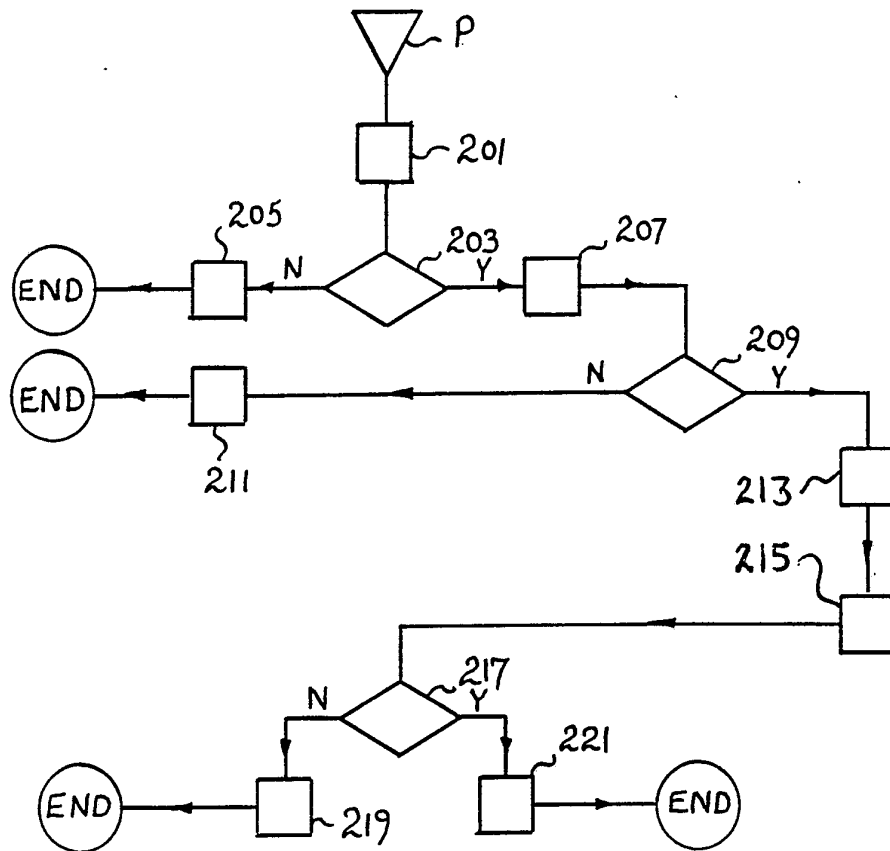


Fig. 5.

**Fig. 6.**

6/18

**Fig. 7.**

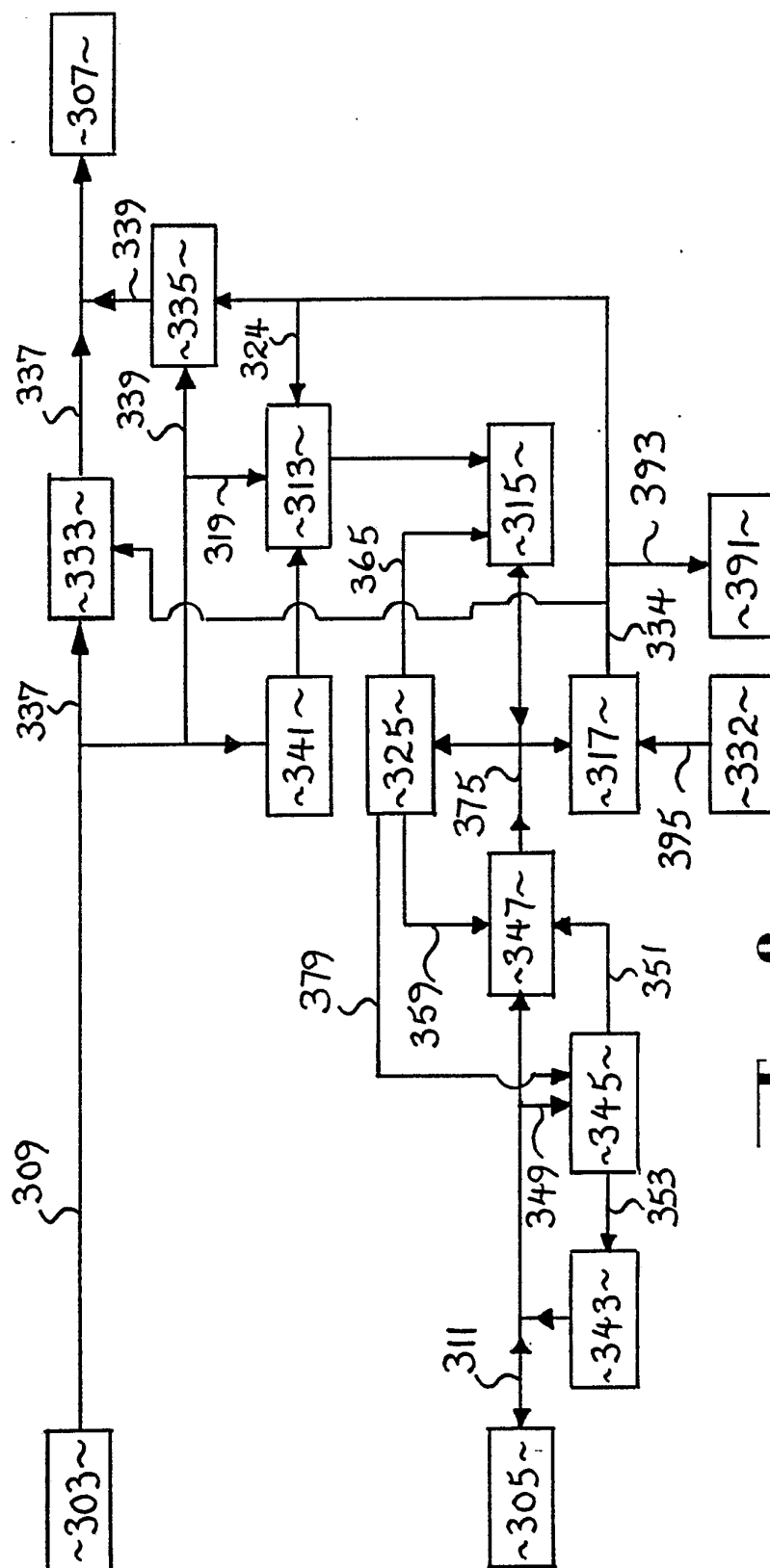


Fig. 8

8/18

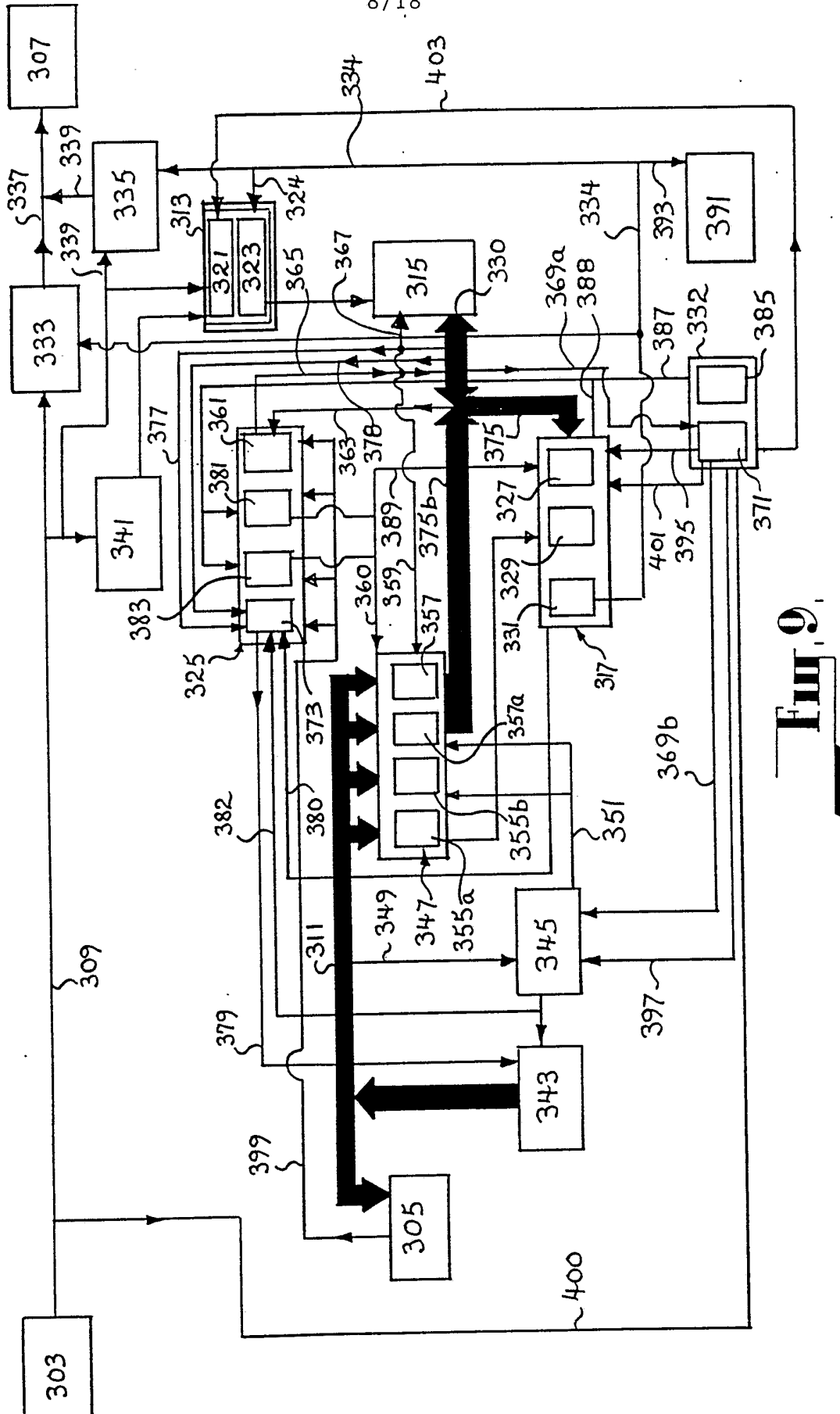


Fig. 9

9/18

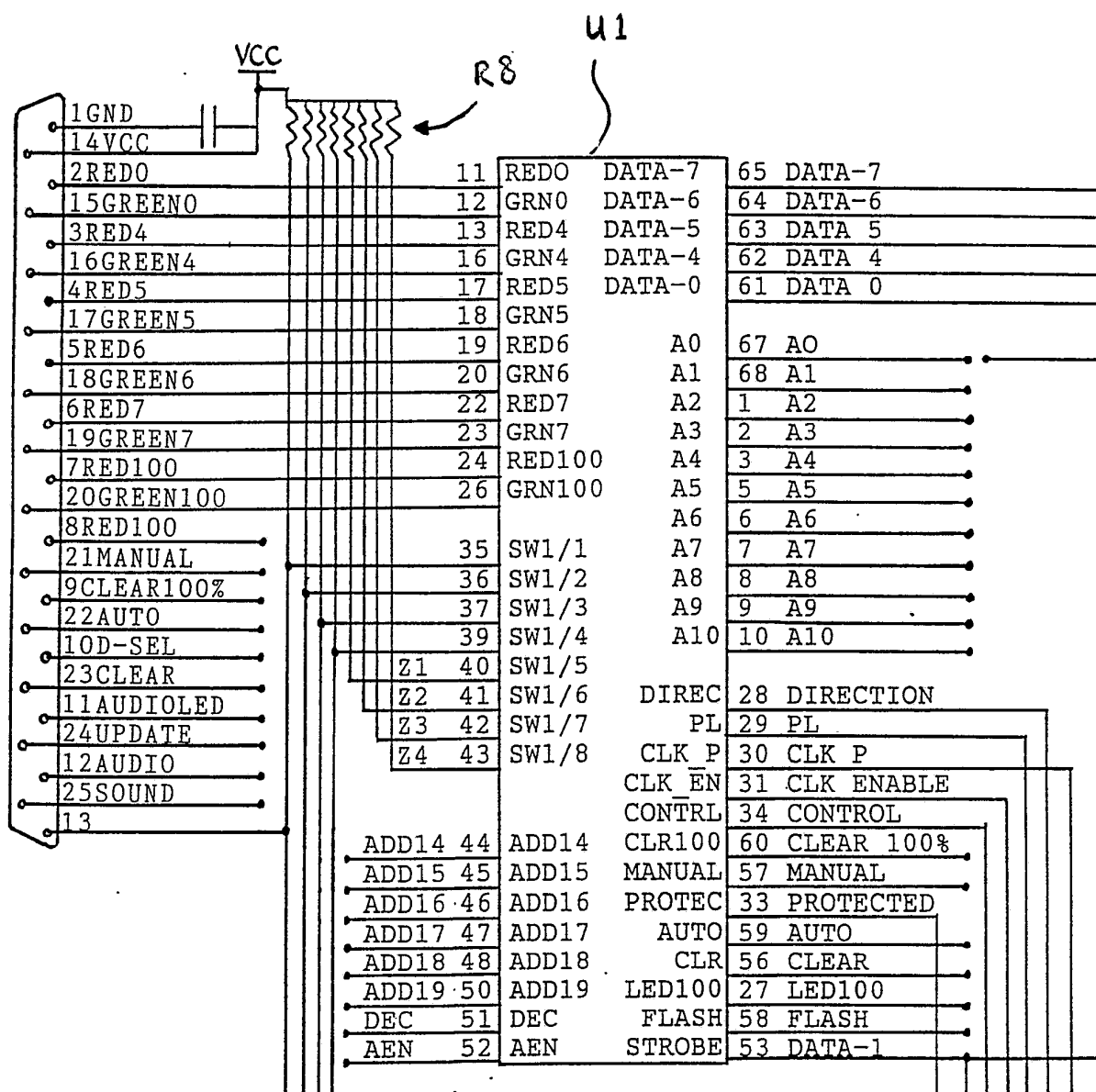


Fig. 10A

10/18

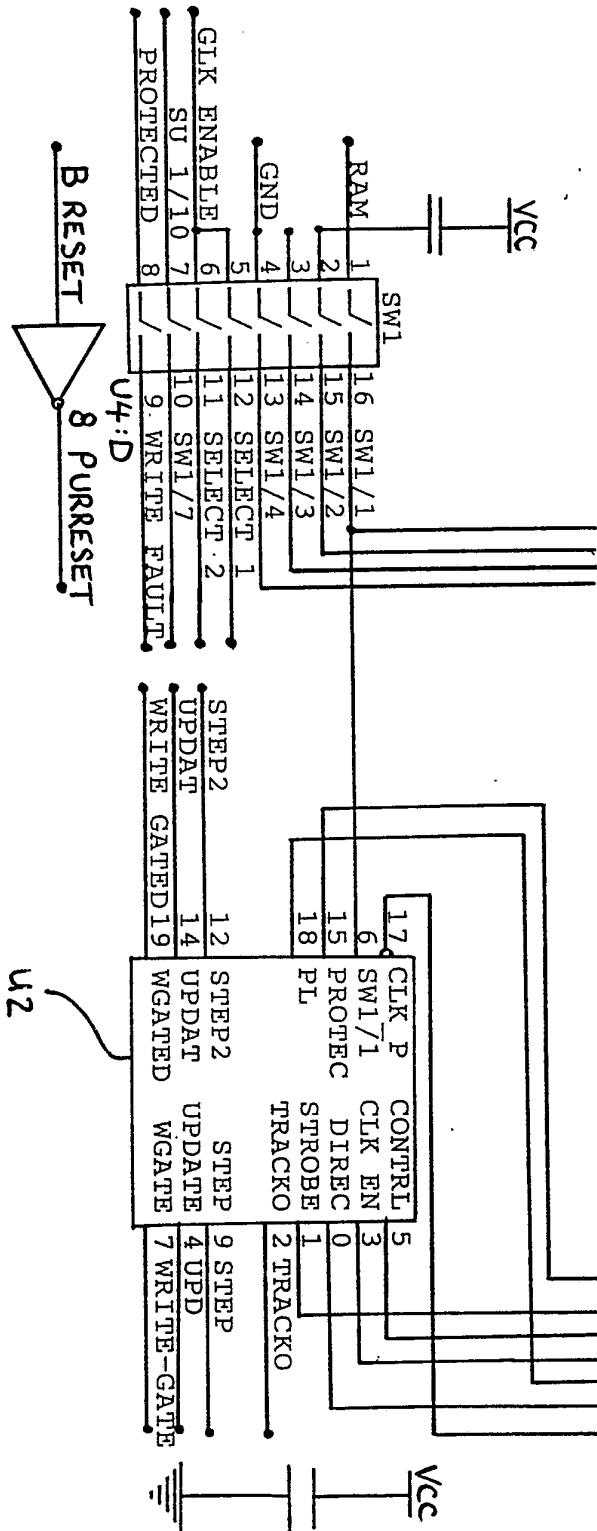
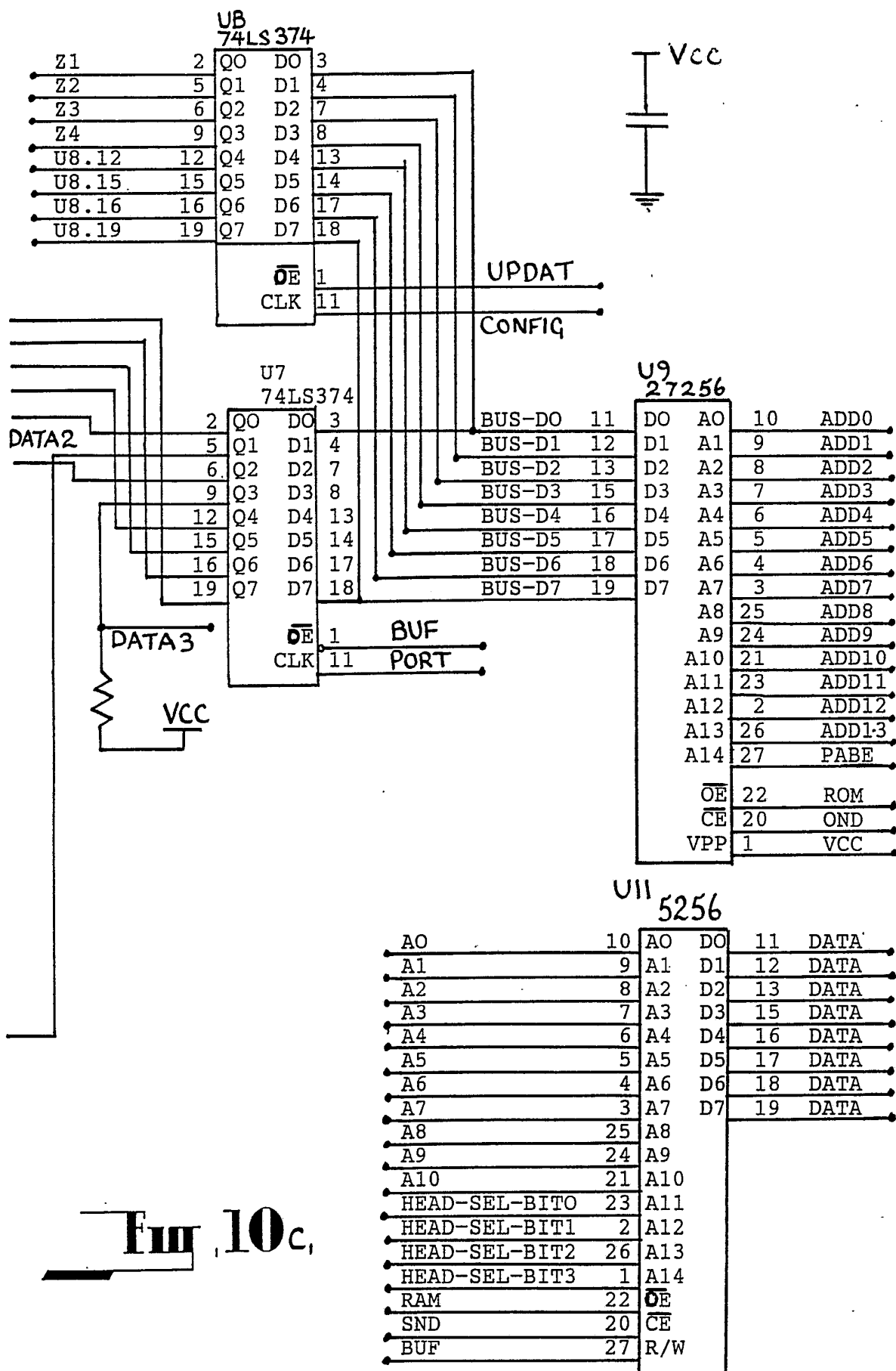


Fig. 10B

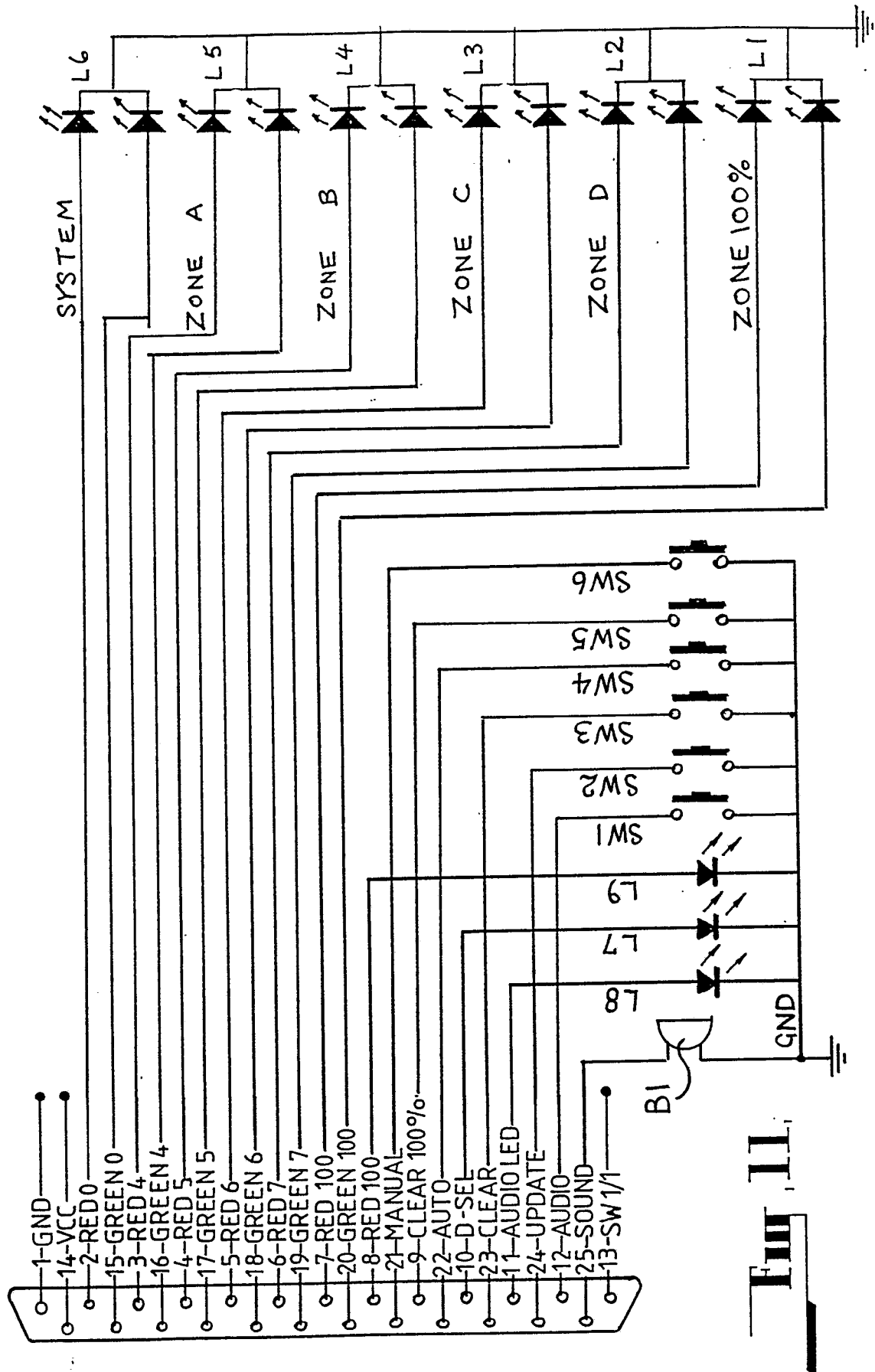
SUBSTITUTE SHEET

11/18

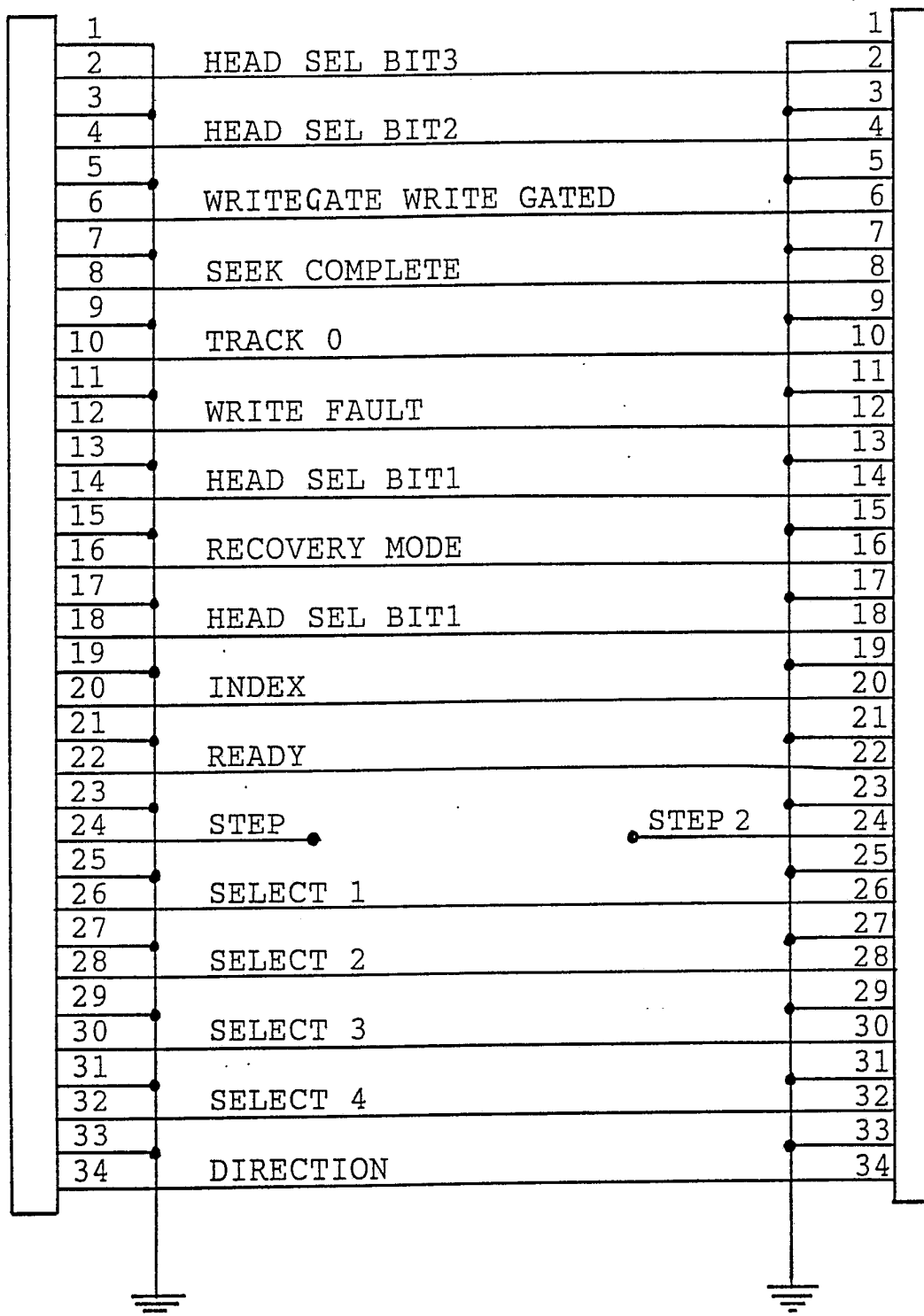


SUBSTITUTE SHEET

12/18



14/18

**Fig. 13.****SUBSTITUTE SHEET**

1	GND	32	IOCHK
2	BRESET	33	BUS-D7
3	VCC	34	BUS-D6
4	IRQ2	36	BUS-D5
5	J10.B5	36	BUS-D4
6	J10.B6	37	BUS-D3
7	J10.B7	38	BUS-D2
8	J10.B8	39	BUS-D1
9	J10.B9	40	BUS-D0
10	GND	41	IOCHRDY
11	MEMW	42	AEN
12	MEMR	43	ADD19
13	IOW	44	ADD18
14	IOR	46	ADD17
15	J10.B15	46	ADD16
16	J10.D16	47	ADD15
17	J10.B17	48	ADD14
18	J10.B18	49	ADD13
19	J10.B19	50	ADD12
20	J10.B20	51	ADD11
21	IRQ7	52	ADD18
22	IRQ6	53	ADD9
23	IRQ5	54	ADD8
24	IRQ4	55	ADD7
25	IRQ3	56	ADD6
26	J10.B26	57	ADD5
27	J10.B27	58	ADD4
28	J10.B28	59	ADD3
29	VCC	60	ADD2
30	J10.B30	61	ADD1
31	GND	62	ADD0

NIOCHK	D7
	D6
	D5
	D4
	D3
	D2
	D1
	D0
	IORDY
	AEN
	ADD19
	ADD18
	ADD17
	ADD16
	ADD15
	ADD14
	ADD13
	ADD12
	ADD11
	ADD10
	ADD9
	ADD8
	ADD7
	ADD6
	ADD5
	ADD4
	ADD3
	ADD2
	ADD1
	ADD0

GND	BRESET
VCC	1RQ2
N/C	DR12
-12V	NZEROS
+12V	GND
MEMW	MEMR
IOW	IOR
NDACK3	DRQ3
NDACK1	DRQ1
NFRESH	4MHZ
IRQ7	IRQ6
IRQ5	IRQ4
IRQ3	NDACK2
TC	BALE
VCC	OSC
GND	

Fig 14

16/18

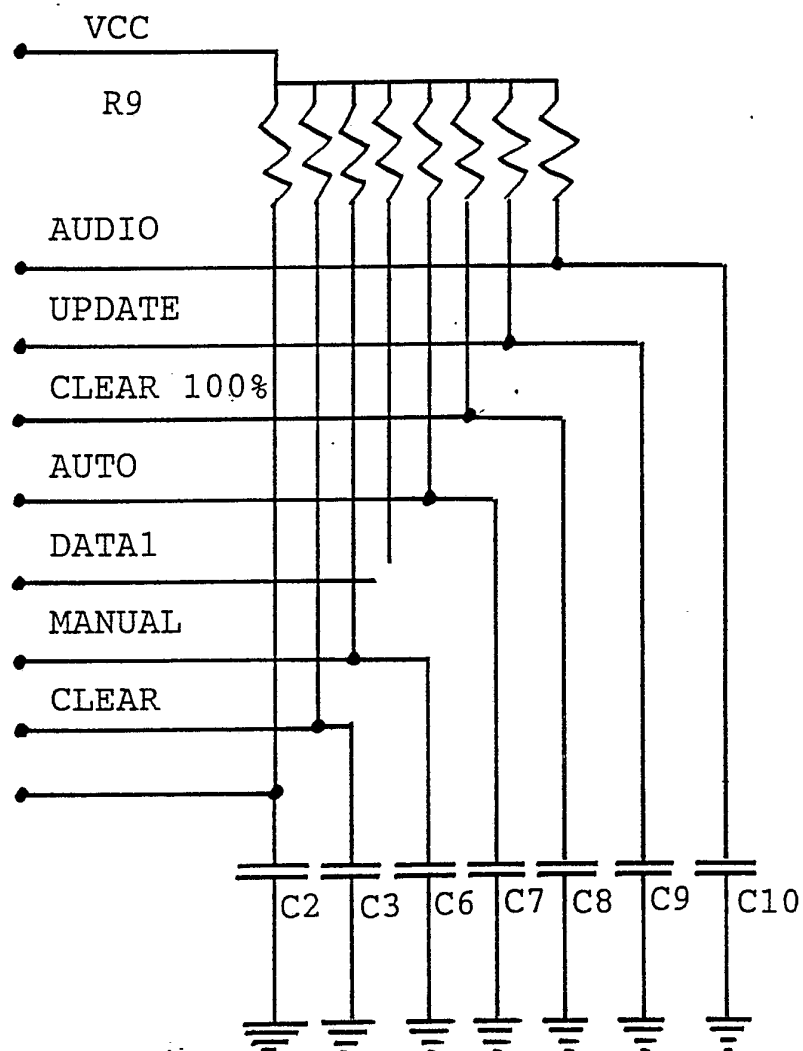
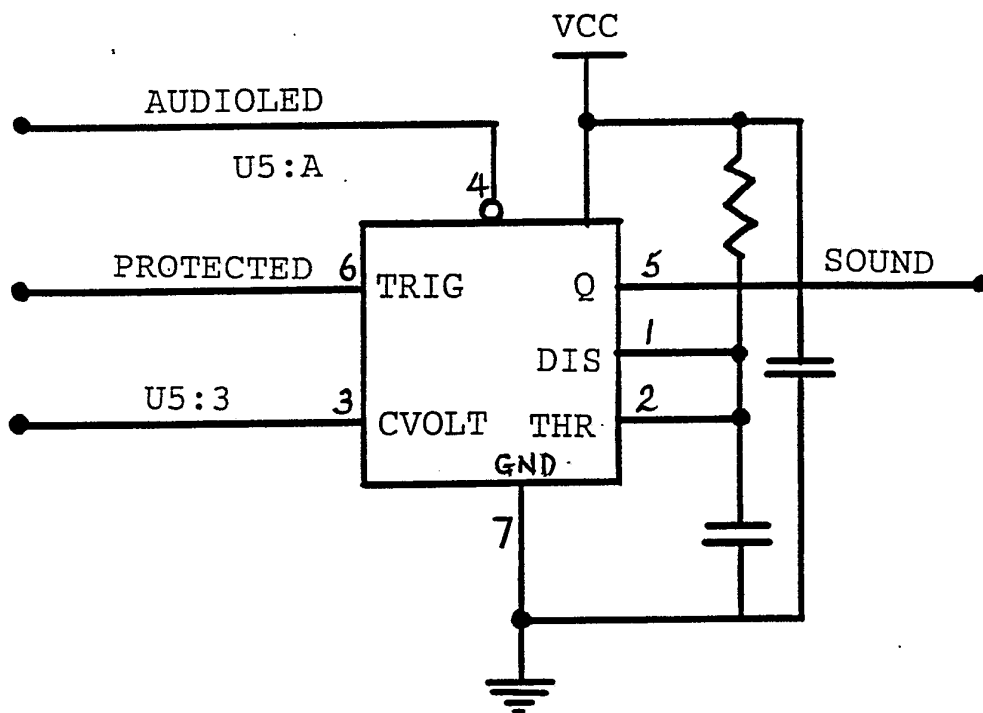
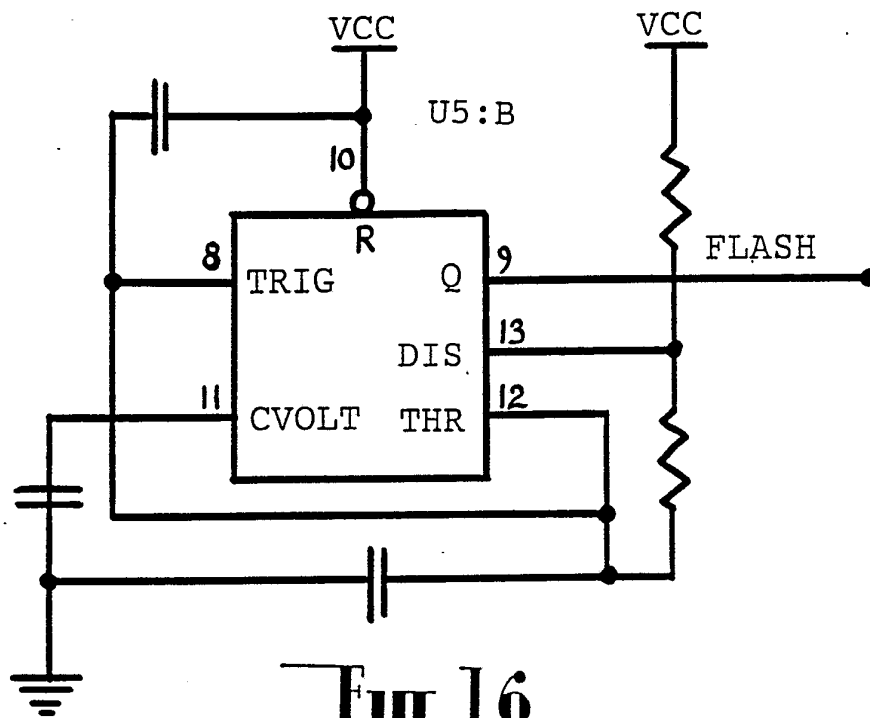
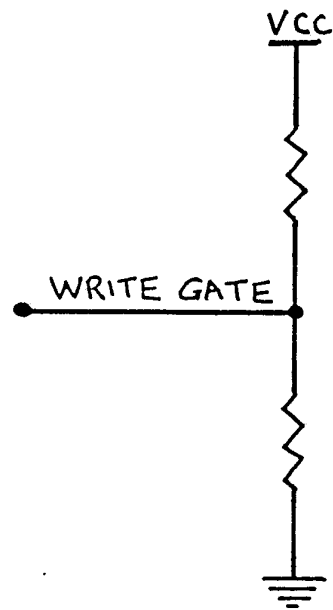
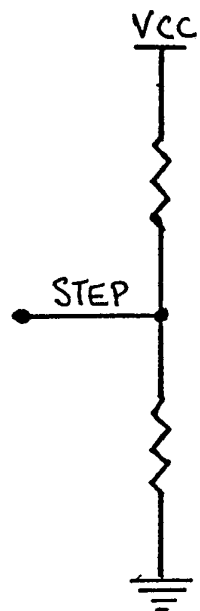


Fig. 15.

17/18



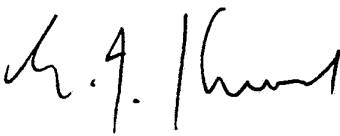
18/18

**Fig. 18.****Fig. 19.**

INTERNATIONAL SEARCH REPORT

International application No.

PCT/AU92/00360

A. CLASSIFICATION OF SUBJECT MATTER Int. Cl. ⁵ G06F 12/14, 12/16, 12/08 According to International Patent Classification (IPC) or to both national classification and IPC				
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) IPC G06F 12/14, 12/16, 12/08 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched AU : IPC as above Electronic data base consulted during the international search (name of data base, and where practicable, search terms used)				
C. DOCUMENTS CONSIDERED TO BE RELEVANT				
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.		
X	S. Walters 'Memory management made easy with the Z 8000', in A. Gupta et al. 'Advanced microprocessors' (1983) IEEE Press	26,28-31,33		
X	G. Martin et al. 'Design considerations of the NS 16082 Memory Management Unit', in A. Gupta et al. 'Advanced microprocessors' (1983) IEEE Press	26,28-31,33		
X	J.F. Stockton 'The M 68451 memory management unit', in A. Gupta et al. 'Advanced microprocessors' (1983) IEEE Press	26,28-31,33		
X	GB,A, 2231418 (S & S ENTERPRISES (AMERSHAM) LIMITED) 14 November 1990 (14.11.90)	43		
A	EP,A, 150522 (DATA GENERAL CORPORATION) 7 August 1985 (07.08.85)	1-50		
<div style="display: flex; justify-content: space-between; align-items: flex-start;"> <div style="width: 45%;"> <input type="checkbox"/> Further documents are listed in the continuation of Box C. </div> <div style="width: 45%;"> <input checked="" type="checkbox"/> See patent family annex. </div> </div>				
<table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top;"> <p>* Special categories of cited documents :</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p> </td> <td style="width: 50%; vertical-align: top;"> <p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p> </td> </tr> </table>			<p>* Special categories of cited documents :</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>
<p>* Special categories of cited documents :</p> <p>"A" document defining the general state of the art which is not considered to be of particular relevance</p> <p>"E" earlier document but published on or after the international filing date</p> <p>"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>"O" document referring to an oral disclosure, use, exhibition or other means</p> <p>"P" document published prior to the international filing date but later than the priority date claimed</p>	<p>"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>"&" document member of the same patent family</p>			
Date of the actual completion of the international search 4 November 1992 (04.11.92)		Date of mailing of the international search report 5 Nov 1992 (05.11.92)		
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200 WODEN ACT 2606 AUSTRALIA Facsimile No. 06 2853929		Authorized officer  E.J. KNOCK Telephone No. (06) 2832206		

Box I Observations where certain claims were found unsearchable (Continuation of Item 1 of first sheet)

This international search report has not established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:
2. ☐ Claim Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:
3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box II Observations where unity of invention is lacking (Continuation of item 2 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

1. ☐ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims
2. ☒ As all searchable claims could be searched without effort justifying an additional fee, this Authority did not invite payment of any additional fee.
3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically
claims Nos.:
4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims;
it is covered by claims Nos.:

Remark on Protest

- ☐ The additional search fees were accompanied by the applicant's protest.
- ☐ No protest accompanied the payment of additional search fees.

This Annex lists the known "A" publication level patent family members relating to the patent documents cited in the above-mentioned international search report. The Australian Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent Document Cited in Search Report		Patent Family Member			
GB	2231418				
EP	150522	BR	8102508	DE	3177242
		EP	149858	JP	57027336
		US	4532590	BR	8102510
END OF ANNEX					