REPLACED SHEETS SG
☑ Description : paged ...1... to paged ..23..
☑ Claim(s) : paged ...1... to paged ...3...
☑ Drawing(s) : paged ...1/11... to paged ...11/11...
☐ Abstract : paged ..... to paged .....

RTMPIS (3) copy

SEIKO EPSON CORPORATION 90/87209 EP

INFORMATION PROCESSING SYSTEM AND SEMICONDUCTOR AUXILIARY STORAGE FOR USE IN THE SYSTEM

Background of the Invention

This invention relates generally to an information processing system of a relatively small size such as a personal computer, and more particularly to an information processing system having a compact and high speed auxiliary storage for storing data and programs.

A conventional information processing system is generally configured as shown in Fig. 14. Such a system comprises a central processing unit (CPU) 1 which is a heart of the system. CPU 1 is connected through system bus 26 to main storage 2 in which programs and data transferred from the auxiliary storage and intermediate results of programs being executed are stored, read-only memory (ROM) 3 which stores a basic input/output system (BIOS), keyboard 4 which functions as an input device to the system, display 5 which functions as an output device, a hard disk 6 and a floppy disk 7 which function as an auxiliary storage to store programs and data.

When power is turned on to the system, the system is reset and initialized and CPU 1 executes an initial load program (ILP) stored in BIOS ROM 3 to transfer the programs from hard disk 6 or the floppy disk 7 to main storage 2. Thereafter, CPU 1 executes the programs transferred to main storage 2 to accomplish the objectives of the programs.

The magnetic storage such as hard disk 6 or floppy disk 7 which are employed as the auxiliary storage has a limit in its possible size reduction, weight reduction and power consumption reduction. As a

This Sheet Has Been Replaced

result, the presence of the magnetic storage imparts a large restriction to the realization of a compact information system featuring actual portability.

Since the prior art floppy disk drive or the hard disk drive includes a mechanical rotary portion and a mechanical operation for moving a read/write head relative to a magnetic medium, access time can be necessarily long and the operation of the drive slow. Further, the mechanical operation involved during the access operation is noisy.

Beside the forgoing disadvantages, it is known that the magnetic storage is relatively weak with respect to a strong magnetic field and a high temperature and humidity environment.

In order to avoid the problems encountered in the magnetic storage, attempts have been made to use a semiconductor storage by following two different approaches. In the first approach, an application program, i.e., a data processing program which allows a user to visually interact with the program and confirm a result, switches banks of the semiconductor auxiliary storage and access is accomplished to the semiconductor auxiliary storage which is deemed as a part of the main storage of the system. In the second approach, the access is made through a device drive program, i.e., a device driver, as a specific auxiliary storage and functions as a part of the operating system (OS).

However, in the first approach, an application program is designed to have a high dependency to particular type and kind of memory employed. An existing application program is usually created and prepared on the assumption that a hard disk or a floppy disk is em-

This Sheet Has Been Replaced

ployed as an auxiliary storage. Thus, it is necessary to modify the application program so that it may be employed with the semiconductor auxiliary storage.

In the second approach, after the device driver is actuated, the semiconductor storage device is set in its activated condition. A BOOT program is stored in a semiconductor device and the main body of software cannot be set up on an active condition. Then, it is not possible to store the OS itself in the semiconductor auxiliary storage, and it is not possible to incorporate a medium copy program, i.e., copied into such a storage, or from a floppy disk to floppy disk and, further, is dependent on a BIOS which has a high dependency on the OS.

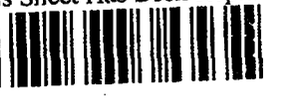
It is an object of the present invention to solve the above mentioned problems.

It is another object of this invention to provide a compact, light weight, highly reliable and high speed information processing system while permitting the utilization of the existing software applications and programs created and written as targeted for use with particular conventional memory means, such as a floppy disk memory or a hard disk memory.

Summary of the Invention

According to this invention, a information processing system of the present invention employes a semiconductor auxiliary storage as an auxiliary storage. A basic input/output program stored in the main body of the system accesses the semiconductor auxiliary storage when it receives an address request for a magnetic auxiliary storage or memory means, which request originates from the execu-

This Sheet Has Been Replaced



tion of a host program. The accessing is accomplished by translating an address for the magnetic storage into an address for the semiconductor auxiliary storage. The address translation is accomplished by utilizing ID information for the magnetic storage which has previously been written into the semiconductor auxiliary storage. The basic input/output program converts the access request for the magnetic storage from an application software or a disk operating system (DOS) to an access request for the semiconductor auxiliary storage and executes the request. An address for the semiconductor auxiliary storage is obtained by reading out the corresponding ID information based on the address for the magnetic storage and putting the ID information and the address for the magnetic storage into a predetermined mathematical formula.

It is preferred that the semiconductor auxiliary storage contains both a read-only storage and a erase/write storage capacity and that the information processing system include means for copying a portion of the content of the read-only storage of the semiconductor auxiliary storage into the erase/write storage. When an access is made to the semiconductor auxiliary storage having the portion of the content of the read-only storage copied into the erase/write storage, it is preferred that the information processing system includes means for translating physical address of a source area to a physical address of a target area. When the portion of the content of the read-only storage of the semiconductor auxiliary storage has been copied into the erase/write storage and an access request is issued to the source area, the access is made by employing a sum of the address and a value determined by the capacity of the copied into area

This Sheet Has Been Replaced



as a physical address. In this manner, the access is made to the target area. Further, when an access is made to the semiconductor auxiliary storage, means may be provided for determining whether the semiconductor auxiliary storage is to be handled as a floppy disk type magnetic storage or a hard disk type magnetic storage based on a determination of the capacity of the semiconductor auxiliary storage.

It is preferable that the semiconductor auxiliary storage be constructed for easy removable mounting in the information processing system.

Other objects and attainments together with a fuller understanding of the invention will become apparent and appreciated by referring to the following description and claims taken in conjunction with the accompanying drawings.

Brief Description of the Drawings

Fig. 1 is a block diagram of one embodiment of an information processing system comprising the present invention.

Fig. 2 is a block diagram of an internal configuration of a semiconductor auxiliary storage employed relative to this invention.

Fig. 3 is a store status of capacity information by memory type in a control register.

Fig. 4 is a hierarchy block diagram of the system configuration for the embodiment shown in Fig. 1.

Fig. 5 is a flow chart of read and write processes of the BIOS to a software interrupt $INT1H_H$ from the DOS.

Fig. 6 is a store status of ID information.

Fig. 7 is a memory space of the semiconductor auxiliary storage.

This Sheet Has Been Replaced

Fig. 8 is a flow chart of read and write processes of the BIOS to the software interrupt $INT1B_H$ from the DOS for selecting a processing procedure based on the capacity of the semiconductor auxiliary storage.

Fig. 9 is a flow chart of a process for determining a capacity for each memory type of semiconductor auxiliary storage.

Fig. 10 is a relationship among a data format, a memory type and a logical address in the memory space of semiconductor auxiliary storage.

Fig. 11 is a flow chart of a copy process.

Fig. 12 is a flow chart of a translation process from a logical address to a physical address.

Fig. 13 is a flow chart of a format track process.

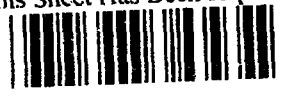
Fig. 14 is a block diagram of a prior art information processing system.

Description of the Preferred Embodiments

Referring now to Fig. 1 there is shown information processing system 1 of this invention comprising CPU 1A, main storage 2, BIOS ROM 3, keyboard 4, display 5, semiconductor storage 9, expansion floppy disk 10 and expansion hard disk 11 all connected to system bus 26. Main storage 2 and BIOS ROM 3 are sometimes referred to collectively as the main storage of system 1 but they are separately referred herein.

CPU 1, main storage 2, keyboard 4, and display 5 are identical to those employed in the conventional information processing system shown in Fig. 14. A physical structure of BIOS ROM 3 is also identical to a conventional BIOS but the content of a program stored therein is

This Sheet Has Been Replaced



different in part because of the necessity to drive and control semiconductor auxiliary storage 9 instead of hard disk 6 and floppy disk 7 in the conventional system. Expansion floppy disk 10 and expansion hard disk 11 may be added as required in order to support the operation of semiconductor auxiliary storage 9. For example, these expansion disks may be added when semiconductor auxiliary storage 9 is not sufficient alone for the memory capacity necessary for auxiliary storage. On the other hand, these expansion disks are not always necessary.

Fig. 2 is a block diagram of an internal configuration of semiconductor auxiliary storage 9. Semiconductor auxiliary storage 9 comprises memory circuit 21 including a semiconductor integrated circuit, data register 22, address register 23 and control register 24. Memory circuit 21 contains two types of memories: a read-only memory (ROM) and an erase/write memory (RAM). Semiconductor auxiliary storage 9 is physically supported on a structure generally referred to as an IC card. It has a card-like plate structure, and when it is plugged into connector 25 to system bus 26, electrical connection is made to system bus 26. In Fig. 2, only one semiconductor auxiliary storage 9 is illustrated for purposes of simplification but a plurality of such storage structures are usually provided so that a large amount of data may be stored among semiconductor auxiliary storages or separate semiconductor auxiliary storages are employed for storing software programs as well as for storing data.

The functional operation of storage 9 is generally as follows. Address register 23 designates an address in memory circuit 21. A start address for access by CPU 1 to memory circuit 21 is set by CPU

This Sheet Has Been Replaced



1 into address register 23 through system bus (I/O bus) 26. After the start address has been set in address register 23, data is written into data register 22 and the data is automatically written into that address in memory circuit 21. When the data is written into data register 22, address register 23 is automatically incremented so that data can be written into the next address. Similarly, when the start address is set in address register 23 and data is read from data register 22, the data is automatically read from that address of memory circuit 21. When the data is read, address register 23 is automatically incremented and data can be read out from the next address.

Control register 24 stores an identification code or device information indicative of the memory type of semiconductor auxiliary storage 9. Fig. 3 shows a store status in control register 24 of the memory capacity by memory type which is a portion of the device information of semiconductor auxiliary storage 9. Fig. 3A shows a type register which indicates the memory type of memory circuit 21. When bit 0 (MKO) of the register is "1", this indicates that the memory type of memory circuit 21 is a RAM. Similarly, bit 1 (MK1) indicates a mask ROM, bit 2 (MK2) indicates a P-ROM, and bit 3 (MK3) indicates an EEPROM.

Fig. 3B shows a capacity register which indicates the erase/write memory capacity of memory circuit 21. When the bit 0 (WCO) of the register is "1", this indicates that the erase/write capacity is 64 kilo bytes (KB). Similarly, bit 1 (WC1) indicates 128 KB, bit 2 (WC2) indicates 256 KB, bit 3 (WC3) indicates 512 KB, bit 4 (WC4) indicates 1

This Sheet Has Been Replaced

MB, bit 5 (WC5) indicates 2 MB, bit 6 (WC6) indicates 4 MB, and bit 7 (WC7) indicates 8 MB.

Fig. 3C shows a capacity register which indicates the read-only memory (ROM) capacity of memory circuit 21. When bit 0 (VC0) of the register is "1", this indicates that the ROM capacity is 64 KB. Similarly, bit 1 (UC1) indicates 128 KB, bit 2 (UC2) indicates 256 KB, bit 3 (UC3) indicates 512 KB, bit 4 (UC4) indicates 1 MB, bit 5 (UC5) indicates 2 MB, bit 6 (UC6) indicates 4 MB, and bit 7 (UC7) indicates 8 MB. The contents of those registers are detected by reading out a specific I/O port address of the information processing system 1 and the memory type and the memory capacity of semiconductor auxiliary storage 9 which is currently loaded can be determined based on the contents of these registers.

As previously indicated, semiconductor auxiliary storage 9 is removably mounted onto system bus 26 through connector 25. Storage 9, therefore, may be exchanged for use relative to each program or data stored in storage 9 as in the case of a conventional floppy disk. Since storage 9 is of a card shape by utilizing semiconductor integrated circuits for memory storage, it is easy to transport and handle. Since a ROM is used entirely or partially in memory circuit 21, one software may be copied in volume as compared to one-to-one copy in the case of a floppy disk. Accordingly, this approach is effective for the distribution of the software.

The operation of information processing system 1 of the present embodiment will now be explained.

Fig. 4 shows a hierarchy block diagram of the system configuration of the system 1 relative to operation in a file access mode.

This Sheet Has Been Replaced



Application software 41 exists at the top level of hierarchy. Next is disk operating system (DOS) 42 which handles software interruptions due to file input/output requests from application software 41. Below DOS 42 is a basic input/output system (BIOS) 43 which is firmware that controls the lower level input/output operations relative to operation of hardware 44. The general configuration of hardware 44 is shown in Fig. 1. CPU 1 may, for example, be an Intel 16 bit microprocessor 8086. DOS 42 may be Microsoft MS-DOS. BIOS 43 may be one of the input/output systems of the disk requested by the software interrupt $INT1B_{II}$ from DOS 42. BIOS 43 is stored in BIOS ROM 3 together with an IPL and it operates under the control of CPU 1. DOS 42 and application software 41 are generally stored in semiconductor auxiliary storage 9 and they are transferred to main storage 2 and executed by CPU 1. DOS 42 may be stored in BIOS ROM 3.

When the information processing system shown in Fig. 1 is started, CPU 1 executes the IPL stored in BIOS ROM 3 and transfers DOS 42 stored in semiconductor auxiliary storage 9 to main storage 2. Then, CPU 1 transfers application software 41 stored in semiconductor auxiliary storage 9 to main storage 2 and the application software is executed under the control of DOS 42.

Application software 41 and DOS 42 are prepared on the assumption that the magnetic storage, such as the floppy disk or the hard disk, is employed as auxiliary storage. Accordingly, BIOS 43 is requested by DOS 42 or application software 41 to access the floppy disk or the hard disk. However, BIOS 43 converts the access request for the magnetic storage to an access request for semiconductor

This Sheet Has Been Replaced



auxiliary storage 9 in the following manner. Accordingly, DOS 42 and application software 41 see the auxiliary storage as the magnetic storage but, in fact, is semiconductor auxiliary storage 9. In other words, in the present system, semiconductor auxiliary storage 9 is treated as if it is the magnetic storage or memory means. As a result, information processing system 1 can make use of existing application software without any need or requirement of changing the content thereof including any necessary adaptation to interface with semiconductor auxiliary storage 9.

Fig. 5 is a flow chart of the read and write processes of BIOS 43 for the software interrupt $INT1B_H$ from DOS 42. When application software 41 issues a file input/output request to DOS 42, DOS 42 determines a drive associated with that file and issues the software interrupt, $INT1B_H$, to BIOS 43. DOS 42 requests BIOS 43 for access to either the floppy disk or the hard disk. BIOS 43 determines the drive associated with the request (step 101), i.e., it determines a drive number and determines if the request is to semiconductor auxiliary storage 9, which is treated as if it were the floppy disk or the hard disk, or to expansion floppy disk 10 or to expansion hard disk 11. If the request is access to expansion floppy disk 10 or expansion hard disk 11, the normal read/write process for the software interrupt $INT1B_H$ is carried (step 106). If the request is access to semiconductor auxiliary storage 9, the content of the ID information stored in semiconductor auxiliary storage 9 is examined to determine if the correct ID information has been formatted (step 102). The ID information is track ID information which would normally be written in a magnetic auxiliary storage or memory means. It is the informa-

This Sheet Has Been Replaced



tion necessary for BIOS 43 to translate the disk address to an address in semiconductor auxiliary storage 9. If the ID information has not been formatted, the address translation cannot be carried out and an error process is carried out (step 105). The ID information will be explained in detail later. If the ID information has been correctly formatted, the disk address (cylinder number, surface number and sector number) is translated into an address in semiconductor auxiliary storage 9 based on the content of the ID information stored in semiconductor auxiliary storage 9 (step 103) and an input/output process is carried out (step 104). The address translation in the step 103 will be explained in detail later.

By the provision of the discrimination function to determine whether BIOS 43 requests the access to semiconductor auxiliary storage 9 or not, and the address translation function and the input/output function for semiconductor auxiliary storage 9 by the reference of the ID information, application software 41 and DOS 42 need not be concerned that semiconductor auxiliary storage 9 is employed as the auxiliary storage rather than a conventional magnetic storage memory means. It can be treated as if the floppy disk or the hard disk is employed as the auxiliary storage.

Referring to Fig. 6, the track ID information for the magnetic storage is now explained. In this particular example, semiconductor auxiliary storage 9 is treated a 640 KB floppy disk. Fig. 6 shows a format of the ID information stored in the first 1 KB area of semiconductor auxiliary storage 9. The ID information includes four bytes, one byte for each cylinder number (C), surface number (H), the number of sectors (R) and the length of sector (N) for each track. The

This Sheet Has Been Replaced



cylinder number (C) is usually written in an ID area in one sector of one track of a floppy disk, and the surface number (H) is also written in the ID area of the floppy disk. The surface number "00" indicates a front surface and "01" indicates a rear surface. The number of sectors (R) indicates the number of sectors in a track. In the present embodiment, the number of sectors is eight for each track. The sector length (N) indicates the sector length in accordance with a sector length code of Table 1. Bit 7 thereof indicates whether it is single density or double density. In the present embodiment, each track is of single density and has 512 bytes.

N	0	1	2	3	4
SECTOR LENGTH	128	256	512	1024	2048

N	5	6	7	8 OR LONGER
SECTOR LENGTH	4096	8192	16384	32768

Table 1

In Fig. 6, numbers in the leftmost column indicate the top address of the ID information of the respective tracks. As many 4-byte ID information for the tracks is provided as the number of tracks of a virtual floppy disk, and entire track ID information 61 having the same data content as that of lost rack ID information 60 is stored in the last four bytes of the top 1 KB of memory. The top address (hexadecimal 27C in the present invention) of last track ID information 60 can be calculated based on entire track ID information 61 by the following formula:

This Sheet Has Been Replaced

(C x 2 x H) x 4.

By letting C = 4F and H = 1 in the above formula, the result is:

(4F x 2 + 1) x 4 = 27C.

The translation from a standard disk address to an address on semiconductor auxiliary storage 9 is now explained. A memory space of semiconductor auxiliary storage 9 is allotted as shown in Fig. 7. The ID information shown in Fig. 6 is stored in the top 1 KB of memory and data is stored in subsequently designated areas in the order of track number.

The translation of the disk address to the address on semiconductor auxiliary storage 9 by BIOS 43 is carried out in the following manner after it has been checked that the ID information stored in the top 1 KB of semiconductor auxiliary storage 9 has been correctly formatted. First, based on parameters Ci and Hi relating to the cylinder number and the surface number which are portions of the disk address delivered from the software interrupt $INT1B_H$, the top address of the ID information of the corresponding track is calculated by the following formula:

(Ci x 2 + Hi) x 4.

For example, by putting Ci = 1 and Hi = 1, which represent the third track into the above formula, we obtain "DOC". Then, based on a parameter Ri relating to the sector number, which is a portion of the disk address delivered from the software interrupt $INT1B_H$, the cylinder number (C), the surface number (H), the number sectors (R) and the sector length (N) are obtained by accessing the top address of the ID information calculated in accordance with the above formula and an input/output address of semiconductor auxiliary storage

This Sheet Has Been Replaced



9 corresponding to the disk address is calculated in accordance with the following formula (1) or (2):

$$400H + \{(C \times 2 + H) \times R + Ri - 1\} \times 128 \times 2^N \quad (1)$$

$$400H + (C \times 2 + H) \times R \times 128 \times 2^N \times 2 + (Ri - 1) \times 128 \times 2^N \quad (2)$$

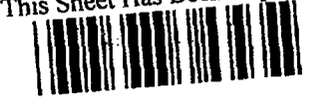
Formula (1) is used in the case of double density while formula (2) is used in the case of single density. The single density and the double density can be discriminated by bit 7 of the sector length, as previously described.

In formulas (1) and (2), 400H (where H represents a hexadecimal number) indicates the top 1 KB in which the ID information is stored, (C x 2 + H) indicates the number of tracks arranged before the track in question on the assumption that there are two surfaces, (C x 2 + H). R indicates a sum of the numbers of sectors arranged before the track in question, and 128×2^N indicates the sector length by bytes in accordance with Table 1. The difference between formula (1) and formula (2) is because the number of sectors (R) for single density is one half of the number of sectors (R) for double density.

In the present embodiment, semiconductor auxiliary storage 9 is employed as a floppy disk. It may also be employed as a hard disk in accordance with the same philosophy. In this case, the ID information for a hard disk is written into semiconductor auxiliary storage 9. When an access request to the hard disk is issued for BIOS 43, BIOS 43 accesses semiconductor auxiliary storage 9 to deliver the fetched data to DOS 42 or application software 41 as the data from the hard disk.

Semiconductor auxiliary storage 9 is removable through connector 25 as explained in connection with Fig. 2. As a result, a semiconduc-

This Sheet Has Been Replaced



tor auxiliary storage of an appropriate capacity may be selectively loaded depending on the application. Usually, the hard disk has a larger capacity than the floppy disk. Thus, semiconductor auxiliary storage 9 which has a relatively large capacity is employed as a hard disk and semiconductor auxiliary storage 9 which has a relatively small capacity is employed as a floppy disk. In processing the software interrupt $INT1B_H$ of the BIOS 43, whether the semiconductor auxiliary storage is to be treated as a hard disk or a floppy disk is determined by the capacity of semiconductor auxiliary storage 9.

Fig. 8 is a flow chart of a process for the software interrupt $INT1B_H$ of BIOS 43. When application software 41 requests a file input/output request to DOS 42, DOS 42 issues the software interrupt $INT1B_H$ to BIOS 43. BIOS 43 determines to which device the request was issued from in order to determine whether or not it is to semiconductor auxiliary storage 9 (step 201). If it is not to semiconductor auxiliary storage 9, then a normal $INT1B_H$ process is carried out (step 205) as is known in the art. If it is to semiconductor auxiliary storage 9, the capacity thereof is examined to determine if it is less than 1 MB or not (step 202). If it is less than 1 MB, an input/output process to treat semiconductor auxiliary storage 9 as the floppy disk is carried out (step 203). The step 203 in Fig. 8 corresponds to the treatment designated in dotted line box 110 of Fig. 5. In step 202, if the capacity is more than 1 MB, an input/output process is carried out to treat semiconductor auxiliary storage 9 as the hard disk (step 204).

Fig. 9 is a flow chart for determining the capacity by memory type of semiconductor auxiliary storage 9 which is required in the step

This Sheet Has Been Replaced

202 in Fig. 8. As a preprocess to determine the memory type, the flags for indicating the memory type, i.e., the ROM flag and the RAM flag are cleared to zero (step 301 and 302). Then, a determination or examination is made as to whether or not the RAM content has been loaded into semiconductor auxiliary storage 9 (step 303). If the RAM has been loaded, the RAM capacity is determined by the erase/write capacity register (step 304), as per the description in connection with Fig. 3B, and the RAM load flag is also set (step 305). The RAM capacity can be calculated by multiplying the content of the erase/write capacity register by 64 KB. Then, the memory type register is again examined to determine whether the ROM content has been loaded into semiconductor auxiliary storage 9 (step 306). If the ROM has been loaded, the capacity of the ROM is determined by the non erase/write capacity register (step 307), as per the description in connection with Fig. 3C, and a ROM load flag is also set (step 308). In this manner, the capacity for each memory type of semiconductor auxiliary storage 9 can be determined.

When memory circuit 21 of semiconductor auxiliary storage 9 is set up to be a ROM only memory, it is not possible to store additional data prepared in information processing system 1. If storage 9 is set up to be a RAM only memory, the stored data is destroyed when a back-up power system, such as a battery, becomes exhausted. Thus, the programs and data which are necessary as a minimum are stored in the ROM, and the programs and data modified during program execution are copied into the RAM during system initialization. When semiconductor auxiliary storage 9 is thereafter employed, the copy area is accessed by translating a logical address to a target physical

This Sheet Has Been Replaced

address. In this way, the minimum required programs and data originally written into storage 9 and modified are not erased and any newly generated data may be additionally stored. This method of operation is now explained in detail.

Figs. 10 illustrates the relationship among the data format, the memory type and the logical address of semiconductor auxiliary storage 9. Fig. 10A is a data format of an OS and is specifically representative of Microsoft MS-DOS. Figs. 10B and 10C show memory types associated with appropriate address locations.

The data format of Fig. 10A comprises spare area 51, block control area 52, entry control area 53 and erase/write data area 54 which should be in the erase/write memory or RAM because its contents may be modified. On the other hand, data area 55, which need only be read, may be in the ROM.

In Fig. 10B, initial programs and initial data of spare area 51, block control area 52, entry control area 53 and erase/write data area 54 have been previously stored in first ROM area 56, and the data of read-only data area 55 has been previously stored in second ROM area 57. ID information has been stored in the top 1 KB of first ROM area 56. The data corresponding to the RAM capacity is transferred from the ROM area to RAM area 58 so that the semiconductor auxiliary storage is initialized. A boundary of first ROM area 56 and second ROM area 57 is determined by the capacity of RAM area 58. The portion of the ROM area which has the same capacity as RAM area 58, which is 128 KB, is allotted from the top of the ROM area starting with first ROM area 56 and the remaining portion is second ROM area 57.

This Sheet Has Been Replaced

The copying is carried out by executing a copy program stored in BIOS ROM 3 via CPU 1. The copy mechanism comprises the copy program, CPU 1 and main storage 2. Fig. 11 shows a flow chart of the copy process. First, the capacity of RAM area 58 is determined (step 401). The method for determining this capacity has been previously described in connection with Fig. 9. When there is no RAM area 58, i.e., when the capacity of RAM area 58 is zero, no transfer process is carried out and the process is terminated. When the capacity of RAM area 58 is determined, an end position of first ROM area 56 is calculated based on the capacity (step 402). Then, the data of first ROM area 56 is copied to RAM area 58 (step 403). Thus, the copy process is completed and semiconductor auxiliary storage 9 is initialized.

Fig. 10C shows the memory type as well as the logical address and the physical address. The term logical address as used therein means an address which indicates a virtual logical space for the information processing system for semiconductor auxiliary storage 9 after initialization via the copying process described above. The physical address means an address which indicates a physical space. Physically, a 4 MB ROM (first and second ROM areas 56 and 57) followed by the 128 KB RAM 58 is arranged, but BIOS 43 is treated as if the 4 MB-128 KB second ROM area 57 exists following the 128 KB RAM area 58, as shown in the logical address column of Fig. 10C.

Fig. 12 is a flow chart of a process for converting a logical address to a physical address when BIOS 43 accesses the initialized semiconductor auxiliary storage 9 in response to a software interrupt $INT1B_H$ to read and write data. First, the disk address is translated to a logical address of semiconductor auxiliary storage 9 (step 404). The

This Sheet Has Been Replaced

method for translating the disk address to the logical address has already been explained in connection with Fig. 5 and it is omitted here. The translated address corresponds to the logical address shown on the right hand of the memory space of Fig. 10C. Then, whether or not the translated logical address is smaller than the RAM capacity (indicated in bytes) is determined (step 405). In the present embodiment example, since the RAM capacity is 128 KB, whether or not the logical address is smaller than 128 KB is determined. In step 405, if the logical address is smaller than the RAM capacity, a sum of the logical address and the ROM capacity (total capacity of the first and second ROM areas) is employed as a real address or the physical address of semiconductor auxiliary storage 9 (step 406). Since the ROM capacity is 4 MB in the present embodiment, 4 MB is added. Accordingly, the logical address actually designates a point in RAM area 58. On the other hand, in step 405, if the logical address exceeds the RAM capacity, the logical address is employed as the physical address in its present form (step 407). This address designates a point in second ROM area 57.

In accordance with this method, the semiconductor auxiliary storage provides for the modification of basic programs and data while they are nonvolatility retained in memory. In the present embodiment, the logical address of RAM area 58 is smaller than the logical address of second ROM area 57, although the logical address of RAM area 58 may be arbitrarily selected. In this case, the translation algorithm from the logical address to the physical address needs to be appropriately modified.

This Sheet Has Been Replaced

Attention is now directed to a format track process which is equivalent to the format track process for the floppy disk and the hard disk. For semiconductor auxiliary storage 9 having memory circuit 21, which is configured as RAM only memory, and in which no information has been written, a format track process must be carried out. Fig. 13 is a flow chart of the format track process for one track. The format track process, however, is sequentially accomplished from the track 0 to the last track. In the case of track 0, when BIOS 43 receives a request for the format track process from DOS 42, BIOS 43 determines the device associated with the request and determines whether the request is to semiconductor auxiliary storage 9 or to other I/O devices, for example, expansion floppy disk 10 or expansion hard disk 11 (step 501). If the request is not to semiconductor auxiliary storage 9, a normal $INT1B_H$ format track process is carried out (step 505). If it is to semiconductor auxiliary storage 9, the address on semiconductor auxiliary storage 9 is calculated based on the disk address for track 0 and an area corresponding to a one-track capacity below the calculated address is filled out by a predetermined data pattern. As an example, the memory space in Fig. 7 is immediately after the ID information area, i.e., address 400. The data pattern, "E5" is usually employed for all words. Thus, the area is filled out by data containing no information. Then, the ID information of the track corresponding to the disk address (disk address indicating track 0) is written into the ID information area (step 503). As an example, the ID information area of Fig. 6 is written into the four-byte area (addresses 000 to 003). Then, the same data as the ID information on track 0 immediately written before is written into

This Sheet Has Been Replaced

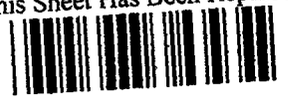


the last four-byte area of the ID information area as the entire track ID information (step 504). With reference to the ID information area of Fig. 6 as an example, the ID information of the track 0 is written into the area starting at the address 3FC. Thus, the format track process of the track 0 is completed and the format track process of track 1 is then carried out. The data area following the data area of track 0 is filled with "E5", the ID information of the first track is written into the four-byte area following the ID information of track 0 and the entire track ID information of the area starting at address 3FC is rewritten as the ID information of the first track. The same process is repeated until the final track is reached and the format track process for the semiconductor auxiliary storage is then complete. In this manner, the DOS and the application program may be handled by semiconductor auxiliary storage 9 in the same manner as they are handled by a hard disk memory or a floppy disk memory.

Since the format track process is carried out in the manner described above, whether or not the formatting is normally done is determined by checking whether the entire track ID information coincides with the last track ID information. If the format process is interrupted and not carried out, normally the entire track ID information does not coincide with the last track ID information.

While the invention has been described in conjunction with specific embodiments, it is evident to those skilled in the art that many further alternatives, modifications and variations will be apparent in light of the forgoing description. For example, Microsoft DOS is employed in the embodiment disclosed although the implementation of the present invention need not be limited to a DOS type system.

This Sheet Has Been Replaced



Further, the present invention is not limited to the BIOS for the purposes of software interrupt $INT1B_H$ but, for example, a disk input/output routine which inputs and outputs to and from semiconductor auxiliary storage 9 may be employed. Thus, the invention described herein is intended to embrace at such alternatives, modifications, applications and variations as fall within the spirit and scope of the appended claims.

This Sheet Has Been Replaced



What is claimed is:

1. An information processing system comprising:

a central processing unit for executing a program;

a main storage for temporarily storing the program to be executed by said central processing unit and data generated by the execution of the program;

semiconductor auxiliary storage means adapted to transfer memory content to and from said main storage; and

a memory for storing a basic input/output program to control hardware of said auxiliary storage means;

said auxiliary storage means being a semiconductor auxiliary storage means;

said basic input/output program being designed to access said auxiliary storage means when an access request to an auxiliary storage means construed as magnetic storage memory means is received as a result of the execution of a host program.

2. The information processing system according to claim 1, wherein said semiconductor auxiliary storage means stores a program including an access request to magnetic storage memory means.

3. The information processing system according to claim 1 or 2, wherein said semiconductor auxiliary storage means has a read-only memory and a erase/write memory, said information processing system including means for copying a portion of content of said read-only memory to said erase/write memory.

4. The information processing system according to claim 1, 2 or 3, wherein said semiconductor auxiliary storage means has a read-only memory and a erase/write memory

This Sheet Has Been Replaced



* C L M S *

and further comprising means for translating a physical address of a source of copy to a physical address of a target of copy when access is made to said semiconductor auxiliary storage means wherein a portion of the content of said read-only memory has been copied to said erase/write memory.

5. The information processing system according to any of claims 1 to 4, wherein access to said semiconductor auxiliary storage means is accomplished by means to translate an address for magnetic storage memory means to an address for said semiconductor auxiliary storage means.

6. The information processing system according to claim 5, wherein said semiconductor auxiliary storage means stores track ID information designated for magnetic storage memory means, said translation means employing said track ID information to make said address translation.

7. The information processing system according to any of the preceding claims, further comprising means for determining whether said semiconductor auxiliary storage means is to be treated as a floppy disk magnetic storage or a hard disk magnetic storage based on the capacity of said semiconductor auxiliary storage means when an access is made to said semiconductor auxiliary storage means.

8. A semiconductor auxiliary storage means employed as a substitute for auxiliary magnetic storage memory means adapted to transfer memory content thereof to and from a main storage of an information processing system thereby permitting the utilization of existing software applications and programs targeted for use with said auxiliary magnetic storage memory means and comprising means to receive an address request for said auxiliary magnetic storage me-

This Sheet Has Been Replaced

* C L M S *

mory means and translate an address for said auxiliary magnetic storage memory means into an address for said semiconductor auxiliary storage means.

9. The information processing system according to any of the preceding claims, wherein said semiconductor auxiliary storage means is removably mounted within said information processing system.

10. A method of utilizing a semiconductor auxiliary storage means in an information processing system while permitting the utilization of previously existing software targeted for use with conventional magnetic storage memory means comprising the steps of:

storing in said semiconductor auxiliary storage means ID information corresponding for said conventional magnetic storage memory means,

capturing an access request for said conventional magnetic storage memory means,

translating an address based on the access request for said conventional magnetic storage memory means to an address based on access to said semiconductor auxiliary storage means employing said ID information, and

thereafter executing the translated request.

This Sheet Has Been Replaced