SYSTEM AND APPARATUS FOR COMPRESSING AND DECOMPRESSING DATA STORED TO A PORTABLE DATA STORAGE DEVICE

CROSS REFERENCE TO RELATED APPLICATIONS

The present application is related to the following United States patent applications, each of which is owned by the assignee of the present invention and is incorporated by reference in its entirety herein:

U.S. Patent Application Serial No. 09/803,173, entitled "PORTABLE DATA STORAGE DEVICE CAPABLE OF BEING DIRECTLY CONNECTED VIA USB PLUG TO A COMPUTER"; U.S. Patent Application Serial No. 09/803,157, entitled "PORTABLE DATA STORAGE DEVICE HAVING SECURE MODE OF OPERATION"; U.S. Patent Application Serial No. 09/898,365, entitled "A PORTABLE DEVICE HAVING BIOMETRICS-BASED AUTHENTICATION CAPABILITIES"; and U.S. Patent Application Serial No. 09/898,310, entitled "A PORTABLE DEVICE HAVING IOMETRICS-BASED AUTHENTICATION CAPABILITIES".

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

The present invention relates to the field of storage of electronic data. More particularly, the present invention relates to the compression and decompression of data stored to a portable memory device.

BACKGROUND OF THE INVENTION

Portable memory devices, sometimes referred to as "key chain" memory devices or Thumbdrives™ (which is a trademark of the assignee of the present invention), are small portable data storage devices. These devices have become a class of indispensable computer peripherals that are widely utilized in business, educational and home computing. These devices are very small in comparison with other data storage devices such as personal computers (PCs), personal digital assistants (PDAs), magnetic disks, or compact disks (CDs). Indeed, the name "keychain" memory device describes the devices as similar in size to a key. Portable memory devices are generally not permanently fitted to a particular host platform,

such as a PC. Rather, they can be conveniently attached to and removed from any computer having the appropriate connection port (e.g., a serial bus port like a USB port, or IEEE 1394 port ("Firewire")). Thus, these portable data storage devices enable a user to transfer data among different computers that are not otherwise connected. Because these devices utilize a non-volatile solid-state memory (e.g., flash memory) as the storage medium, they do not require moving parts or a mechanical drive mechanism for accessing data. The absence of a drive mechanism enables portable data storage devices to be more compact than surface storage devices such as magnetic disks and CDs. Also, because there are no moving parts, reading and writing to the memory can be done much more rapidly than to magnetic disks and CDs. Portable data storage devices also have a much higher memory capacity than magnetic disks, holding up to 256 megabytes, as compared to 1.4 megabytes for magnetic disks.

Although storing data to a portable memory device has proven useful, the utility of these devices is limited by their inability to store larger quantities of data without increasing their size.

SUMMARY OF THE INVENTION

Accordingly, it is an object of the present invention to provide a system and apparatus that enables users to save larger quantities of data to a portable memory device than the current state of the art would allow. This would allow manufacturers to increase the amount of data that can be stored to the memory of these devices without increasing the physical size of the memory and thus the size of the devices. Additionally, this would have the added benefit of allowing manufacturers to decrease the size of these devices without decreasing the amount of data that such devices are capable of storing.

According to one aspect of this invention, there is provided a unitary portable data storage device as defined in claim 1 of the appended claims.

According to another aspect of this invention, there is provided a method as defined in claim 7.

According to a further aspect of this invention, there is provided a method as defined in claim 12.

Preferably, the device is capable of easy connection to a host platform such as a PC via a USB port, IEEE 1394 (i.e., firewire) or similar port. Included in the device may be flash memory, read-only-memory (ROM), random-access-memory (RAM), USB controller, and a micro controller. In addition, a data compression and decompression engine is included in the portable memory device to compress/decompress data as it is stored and retrieved from memory.

The portable memory device (Thumbdrive ™) is connected to a host platform such as a personal computer (PC) by a USB connector inserted into the PC's USB port. In one embodiment, data to be saved to the portable memory device is transmitted by the PC in the form of data packets to a working buffer in the portable memory device. The compression engine retrieves the data from the buffer and compresses each packet of data. Compressed data packets are stored in the flash memory. To retrieve data, the compression/decompression engine retrieves the data from the flash memory and decompresses it. The decompression engine places the decompressed data in the working buffer where it is then transferred to the host platform.

Preferably, the compression/decompression software is located in the ROM then loaded into micro controller for execution. In an alternative embodiment, the compression/decompression engine resides in an ASIC chip in the portable device.

The compression/decompression features of the portable device are invisible to the user. The user saves data but is not aware that it is being compressed. In an alternative embodiment, the user is aware of these features and enables/disables the compression feature as desired. Disabling the compression feature has the effect of increasing the speed at which the data can be saved.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features, and advantages of the present invention will be apparent from the following detailed description of the preferred embodiment of the invention with references to the following drawings.

Fig. 1 is a block diagram illustrating functional blocks of one embodiment of the portable memory device according to the present invention and an illustrative configuration thereof.

Fig. 2. illustrates a flow diagram that describes how the host platform communicates commands to the portable memory device.

Fig. 3. illustrates a flow diagram that describes how data is transmitted, compressed and saved to the memory of the portable memory device.

Fig. 4. illustrates a flow diagram that describes how compressed data saved to the memory of a portable memory device is accessed.

DETAILED DESCRIPTION OF THE INVENTION

The present invention will now be described more fully with reference to the accompanying drawings, in which preferred embodiments of the invention are shown. The present invention may, however, be embodied in many different forms and should not be construed as being limited to the embodiments set forth herein; rather these embodiments are provided so that this disclosure is thorough and complete and fully conveys the invention to those skilled in the art. Indeed, the invention is intended to cover alternatives, modifications and equivalents of these embodiments, which are included within the scope and spirit of the invention as defined by the appended claims. Furthermore, in the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it is clear to those of ordinary skill in the art that the present invention may be practiced without such specific details. In other instances, well known methods, procedures, components, and circuits have not been described in detail so as not to unnecessarily obscure aspects of the present invention.

Figure 1 is a block diagram illustrating functional blocks of one embodiment of the portable memory device and system of the present invention and an illustrative configuration thereof. **Figure 1** shows a portable memory device **100** (a small handheld device capable of fitting substantially in a closed fist) that is connected via a universal serial bus (USB) port IEEE 1394 (i.e., firewire) or similar port connector **40** to a USB port (or similar port) **192** of the host platform **195** (such as a personal computer). In the preferred embodiment, the connection is made without wire or cable interconnection. Host platform **195** is coupled to a power supply circuit **150** located in portable device **100**. Power supply circuit **150** draws power from host platform **195** and serves as a power source for various components of portable device **100**.

Referring still to **Figure 1**, the portable memory device **100** includes a micro controller **160**, flash memory **180**, and a data compression/decompression engine **190**. It is understood that micro controller **160** could also be a general purpose microprocessor.

In one embodiment, a working buffer **165** is located in the micro controller. In an alternative embodiment, the working buffer **165** is located outside of the micro controller **165** (e.g., in the RAM **110**). In the preferred embodiment, the working buffer has 16 kilobytes of capacity.

It should be appreciated that data compression/decompression engine **190** maybe implemented in various ways within the scope of the present invention. In one embodiment, compression/decompression engine **190** is implemented as firmware stored in a non-volatile memory **110** within portable device **100**. In another embodiment, data compression/decompression engine **190** is implemented as part of micro controller **160**. In still another embodiment, compression/decompression engine **190** is implemented as a processor separate from micro controller **160**. In other words, within the scope of the present invention, compression/decompression engine **190** is not required to reside in a particular location in the portable device **100**. Instead, where compression/decompression engine **190** is to be placed is a design choice, thus affording design flexibility to suit different applications in which the present invention can be utilized.

Data compression/decompression engine **190** compresses data after the data is transferred from the host platform **195** to the portable memory device **100.** The data is transferred in the form of data packets passed along a USB bus. The data compression/decompression engine **190** retrieves the data from the working buffer **165** (typically one sector at a time) and compresses the data. In the preferred embodiment, the data is compressed to 50% of its original size. Thus a packet of 512 bytes of data will be compressed to 256 bytes. After compression, the data is stored to the flash memory **180.** The compression/decompression engine **190** also decompresses data stored to the flash memory **180** and transfers the data to the host platform **195.**

In this embodiment, the portable device includes a USB device controller **130,** which serves to control the communication between portable device **100** and host platform **195,** such as a USB-compatible PC having a USB host controller **197** therein.

With reference still to **Figure 1,** portable device **100** also includes a volatile memory **120** and a non-volatile memory **110**. In a preferred embodiment, volatile memory **120** is a random access memory (RAM) that serves as a working memory for micro controller **160**

during its operation. Non-volatile memory **110** is a read-only memory (ROM) in this embodiment and can be used to store firmware that performs various functions of portable memory device **100**.

Additionally, in this embodiment, portable device **100** includes a write-protect switch **170** which, when activated, triggers micro controller **160** to disable write-access to flash memory **180**. It should be appreciated that write-protect switch **170**, like compression/decompression engine **190**, may be implemented in various ways within the scope of the present invention. For example, write-protect switch **170** can be implemented by software (e.g., firmware stored in a non-volatile memory), as part of micro controller **160**, or as a processor unit separate from micro controller **160**. In an alternative embodiment, the write-protect switch **170** can be a manual switch which is manually operated by the user.

In a currently preferred embodiment as illustrated in **Figure 1**, microprocessor **160** controls various components of portable device **100**, including USB device controller **130**, ROM **110**, and RAM **120** (and execution of firmware code stored therein).

Optionally, portable device **100** further includes a USB port (not shown) that is *provided as a convenient feature* that can be used to couple other USB-compatible device(s) to the portable device **100** via USB.

In one embodiment, driver software (not shown), application programming interface (API) (not shown) and monitoring software (not shown) reside in the host platform **195** and communicate with USB host controller **197** to facilitate the operation of portable device **100**.

The architecture of a portable data storage device is described in U.S. Patent application entitled "A Portable Device Having Biometrics-Based Authentication Capabilities" with Ser. No. 09/898,365, filed on July 3, 2001 and in U.S. Patent Application entitled "Portable Data Storage Device Capable of Being Directly Connected Via USB Plug to a Computer" with Serial No. 09/803,173.

Figure 2 illustrates a flow diagram of the operation that occurs when the user connects the portable device **100** to the host platform **195** (step 205) and initiates the saving of data to the portable device **100**. In step 205, upon being coupled to a host platform **195**, portable device **100** undergoes an initialization procedure. In a currently preferred

embodiment, the initialization procedure involves establishing communication with the host platform **195** and ensuring the host platform **195** is aware that portable device **100** has been coupled thereto. Host platform **195** requests services from portable device **100** by sending request packets to USB host controller **197**. USB host controller **197** transmits packets through USB port **192** to USB connector **140**. Portable device **100** then performs various operations such as reading, writing or erasing coupled with compressing and decompressing data from or to flash memory **180**.

In step 210, host platform **195** may send a request to portable memory device **100** to write data to flash memory **180** in the form of a write command. In step 215, USB controller **130** receives write command. Write command typically indicates the number of packets of data to follow that are to be saved.

Figure 3 illustrates a flow diagram of the operation that occurs when data is transferred to the portable device to be compressed and written to the flash memory **180**. Host platform **195** transmits a first data packet to USB controller **130** (step 303). In one embodiment, in accordance with the USB 2.0 standard, each data packet is the equivalent of one sector of memory or 512 bytes. In an alternative embodiment, in accordance with USB standard 1.1, each data packet is 64 bytes.

USB Controller **130** receives packet of data from host platform **195** (step 305). USB controller **130** transmits data packet to working buffer **165** (step 310). In the preferred embodiment, working buffer **165** can hold a maximum of 16 kilobytes of data. If working buffer **165** reaches maximum capacity and can no longer store additional data, then the next data packet may be rejected and a negative acknowledgement (NAK) communication is sent by micro controller **160** to host controller **195** indicating that the rejected data packet was not received (steps 312 and 313). Host controller **195** then retransmits rejected data packet (step 303). This process of packet rejection and retransmission continues until the working buffer **165** has capacity to accept an additional data packet (step 314).

Micro controller **160** executes data compression/decompression engine **190** (Step 315). Compression/decompression engine **190** operates on packets of data stored in working buffer **165** (Step 320) in a First In/First Out scheme whereby data packets are compressed in the order that they are stored to working buffer **165** with the first packet stored in working buffer **165** being the first to be compressed and transferred to flash memory **180**. In the

preferred embodiment, data is compressed to 50% of its original size. In the preferred embodiment, when operating on a USB 2.0 standard, compression engine **190** operates to compress one sector (512 bytes) of data at a time. In an alternative embodiment, when operating on the USB 1.1 standard, compression engine **190** operates to compress 64 bytes of data at a time. Compressed data is saved to flash memory **180** (Step 325).

Memory mapping is controlled by the micro controller **160**. In the preferred embodiment, the host platform (and user) are unaware that the data is being compressed. To the host platform, data appears to fill sectors in the flash memory **180** as if the data were not compressed (i.e., 512 bytes of non-compressed data per sector). Thus compressed data written to the first sector would appear to the host controller to fill one entire sector. In reality, the data is compressed from 512 to 256 bytes and is saved to half the sector space that would be needed for the data had it not been compressed. Thus the second packet of data, after it is compressed from 512 bytes to 256 bytes, may also be saved to the first sector. To the host controller, the second data packet, which in reality is saved to the first sector, will appear to be saved to the second sector. Similarly, the first data packet saved to second sector will appear to be saved to the third sector and the second data packet saved to the second sector will appear to be saved to the fourth sector.

Typically standard 2 bit error detection and 1 bit correction mechanisms are employed.

If there is another packet of data, then host controller **195** transmits the packet of data to USB controller **130** (step 335 and 303). Data compression engine **190** continues to compress additional packets of data stored to working buffer **165** (step 320). When there are no more data packets to be compressed, then the data compression and storage session ends.

Figure 4 illustrates the steps by which data is retrieved from the memory 180 of the portable memory device **100** and decompressed.

In step 410, user connects the portable device **100** to a host platform **195**. In step 410, upon being coupled to a host platform **195**, portable device **100** may undergo the initialization procedure described previously. Host platform **195** requests services from portable device **100** by sending request packets to USB host controller **197**. USB host controller **197** transmits packets through USB port **192** to USB connector **140**.

Host platform **195** may send a request to portable device **100** to read data requested by user (step 420) in the form of a read command. USB controller **130** receives the read command (step 425). Read command indicates the location of the data in the flash memory **180** and number of packets of data to be read. Because host platform **195** is unaware that data is compressed, the read command will request data that is not necessarily in the sector in which the host platform **195** believes it to be. For example, as discussed previously, the second data packet, which in reality is saved to the second half of the first sector, will appear to the host controller **195** be saved to the second sector. Similarly, the first data packet saved to second sector will appear to be saved to the third sector and the second data packet saved to the second sector will appear to be saved to the fourth sector. Thus, host platform's **195** data request may indicate that a particular packet of data is stored to the second sector, where in reality, it is stored to the second half of the first sector. To account for this, micro controller **160** operates to retrieve requested data from flash memory **180** in accordance with the manner in which it was saved (step 325).

In step 430, micro controller **160** executes data compression/decompression engine **190**. Compression/decompression engine **190** retrieves packets of requested data from flash memory **180** (step 445). Compression/decompression engine **190** acts to decompress dates (step 447). Decompressed data is transferred to working buffer **165** (step 450). Decompressed data is transferred to USB controller **130** (step 455). In step 460, data packet is transferred to the memory (e.g., RAM) of the host platform **195**.

Data compression/decompression engine **190** continues to retrieve and decompress data from flash memory **180** until all data requested has been decompressed and transferred to host platform **195**.

Alternatively, host platform **195** may send a request to portable device **100** to read directory of data stored to flash memory **180**. User can then view directory on host platform **195** to determine which data user wishes to retrieve from flash memory **180**.

It is to be understood that the above description is only of the preferred embodiments of the invention. One skilled in the art may devise numerous other arrangements without departing from the scope of protection conferred by the claims.

CLAIMS

1. A unitary portable data storage device having

a universal serial bus (USB) connector adapted to the storage device, the device capable of being directly connected to a computer via the connector to receive data in the form of data packets, the storage device including:

a flash memory having a first storage capacity; and

a data compression engine,

the data compression engine capable of compressing each of the data packets to form respective compressed data in the form of compressed data packets for storage in the flash memory so that the capacity of the flash memory exceeds the first storage capacity.

2. The unitary portable data storage device according to claim 1, further comprising a data decompression engine capable of decompressing compressed data stored to the memory of the portable device.

3. The unitary portable device according to claim 1 wherein a user thereof is not aware that the data is being compressed.

4. The unitary portable device according to claim 1 wherein each of the data packets is compressed to 50 percent of its original size.

5. The unitary portable device according to claim 1 wherein the capacity of the memory is increased by 100% as a result of compressing the data packets into compressed data packets.

6. The unitary portable device according to claim 1 and arranged such that a user thereof can enable and disable the compression engine.

7. A method of compressing data, the method comprising the steps of:

(a) directly connecting a portable data storage device to the communication port of a host platform via a universal serial bus connector;

(b) receiving data from the communication port as a stream of data packets;

(c) executing a data compression engine residing on the portable storage device;

(d) compressing the data packets provided by host to form respective decompressed data packets; and

(e) saving the compressed data packets to the memory of the portable device.

8. A method according to claim 7 wherein the user thereof is not aware that the data is being compressed.

9. A method according to claim 7 wherein the compressing step comprises compressing each of the data packets to 50 percent of its original size.

10. A method according to claim 7 wherein the compressing step results in increasing the capacity of the memory by 100%.

11. A method according to claim 7 and including the additional step of enabling or disabling the compression engine.

12. A method of decompressing compressed data packets saved to the memory of the portable memory device, the method comprising the steps of:

(a) connecting the portable data storage device directly to a communication port of the personal computer via a universal serial bus (USB);

(b) executing a data compression/decompression engine residing on the portable storage device; and

(c) retrieving the compressed data packets from the memory of the portable data storage device;

(d) decompressing the data packets to form respective decompressed data packets; and

(e) transmitting the decompressed data packets to the host platform.

13. The method as in claim 12 and including the additional step of enabling or disabling the compression engine.